MANAGEMENT & DATASCIENCE

# COMMENT UTILISER LE MACHINE LEARNING POUR GAGNER DES MARCHÉS PUBLICS ?

JOEL TANKEU, PHILIPPE ADIABA, STEVE ELANGA & NADIA TOUATI

Publié dans Management & Datascience Vol.4 N°6, le 21 novembre 2020
Catégorie : Application
https://management-datascience.org/articles/14442/

RÉSUMÉ

Nous vivons dans un monde de plus en plus gouverné par la donnée, où les limites des possibilités et des applications ne cessent d'être repoussés jours après jours. Notamment en ce qui concerne les marchés publics, il est possible en exploitant la puissance du machine learning de dégager un avantage concurrentiel considérable. Ainsi, nous développons au cours de cette analyse réalisée dans le cadre d'un data challenge [https://management-datascience.org/challenges/predire-une-decision-dans-les-marches-publics-europeens-2/], la méthodologie que nous avons utilisée afin de prédire non seulement le nombre de candidature à un appel d'offre, mais également le montant du marché qui sera attribué

DATASCIENCE | FORÊTS ALÉATOIRES | MARCHÉS PUBLICS

**Citation :** TANKEU, J., ADIABA, P., ELANGA, S., & TOUATI, N. (Nov 2020). Comment utiliser le machine learning pour gagner des marchés publics ?. *Management et Datascience*, *4*(6). https://management-datascience.org/articles/14442/.

**Les auteurs :**

- **Joel TANKEU**
  - (Pas d'affiliation)
- **Philippe ADIABA**
  - (Pas d'affiliation)
- **Steve ELANGA**
  - (Pas d'affiliation)
- **Nadia TOUATI**
  - (Pas d'affiliation)



**Liens d'intérêts :** Le ou les auteurs déclarent ne pas avoir connaissance de conflit d'intérêts impliqués par l'écriture de cet article.

**Financement :** Le ou les auteurs déclarent ne pas avoir bénéficié de financement pour le travail mis en jeu par cet article.

TEXTE COMPLET

## Introduction

Dans une ère de révolution numérique de plus en plus orienté Data driven, l'intérêt du machine Learning n'est plus à démontrer. Avec un environnement professionnel en perpétuel évolution, où chaque détail compte dans le processus décisionnel, les algorithmes de machine Learning peuvent permettre de dégager un avantage stratégique capable de faire la différence.

En l'occurrence, les marchés publics représentent un cas d'étude intéressant, où l'impact d'outils d'aide à la décision performant est plus que déterminant. Dans l'environnement spécifique européen où environ 90% des procédures de passations de marchés publics sont ouvertes, cela élargit le champ des possibilités pour les fournisseurs et accrus la concurrence. Le processus d'appel d'offre reposant essentiellement sur un système d'enchère, Il est donc primordial pour ces fournisseurs de décider très rapidement s'ils doivent se positionner ou non sur un appel d'offre. Et si oui de faire la proposition juste de devis qui assurera l'équilibre parfait entre la

rentabilité du marché et la satisfaction du donneur d'ordre. Dès lors, il est aisé de comprendre la mine d'or que représenterait pour un manager la capacité de prédire le nombre de candidatures qu'il y aura à un appel d'offre, ou alors le montant du marché qui sera attribué. Découvrir cette mine d'or est le défi gargantuesque auquel nous nous attaquons tout au long cette analyse.

# Données brutes

Le jeu de données utilisé pour cette étude est issu de la commande publique de l'Union Européenne et publié au **Tenders Electronic Daily (TED) [https://ted.europa.eu/TED/browse/browseByMap.do]**. Les données décrivent des marchés publics de l'espace économique européen, la Suisse et l'ancienne République yougoslave de Macédoine pour une période allant de 2009 à 2016.

Le jeu d'entraînement compte plus de deux millions de lignes et le jeu de test environ un million. Chaque instance comporte 18 variables qui définissent des informations sur les contrats (type de contrat, secteur d'activité, etc…), les attributions de marché (la procédure d'attribution, les critères d'attribution, etc…) et le type d'organisme public qui commande le marché. Les élément cibles à prédire sont, le nombre d'offres reçues (NUMBER_OFFERS) et la valeur totale du marché sans TVA en euro (AWARD_VALUE_EUR).

Le jeu de données est constitué de 13 variables catégorielles et 5 valeurs numériques, et on note 4996 lignes dupliquées. 13 variables présentent des valeurs manquantes notamment « B_DYN_PURCH_SYST » et « B_CONTRACTOR_SME » avec respectivement 86.44 % et 85.53 % de valeur manquantes

# Préparation des données

Après une analyse préliminaire du jeu de donnée, nous avons pu remarquer que le jeu de donnée comportait de nombreuses impuretés. D'abord en termes de valeur manquantes ou mal renseignés, mais également du fait des lignes dupliquées. Ces dernières sont pour la plupart dû au fait que certains marchés sont subdivisés en sous-appels d'offres qui ont par conséquent les mêmes caractéristiques mais avec une valeur de marché en sortie différentes. Il est pour le moment impossible de se prononcer sur l'indépendance et la pertinence de l'ensemble des variables.

Nous avons donc effectué un travail considérable de nettoyage de la donnée, à travers un ensemble d'opérations consigné dans un pipeline.

## Suppression des variables inexploitables

Nous avons tout d'abord exploré pour l'ensemble des variables la distribution des valeurs possibles. L'objectif ici était de détecter s'il est possible d'appliquer des transformations sur ces valeurs dans l'objectif de simplifier le modèle. Cette démarche nous a conduit à porter particulièrement notre attention sur la colonne « *MAIN_ACTIVITY »* qui possède 449 valeurs uniques. Un affichage des valeurs de cette colonne nous a permis de constater que chaque ligne de cette colonne était une énumération des différents secteurs d'activités concernés par le marché, chaque valeur étant séparé par une virgule (*value1, value2, …, value n).* Dans un premier temps nous avons opté pour transformer cette énumération, en tableau de valeurs en se basant sur le séparateur «, », mais il s'est avéré que la donnée n'était pas correctement renseignée, notamment des virgules mal positionnées dissimulées dans plusieurs énumérations. Ce qui faussait la séparation des valeurs. (Par exemple, au lieu de « value1,value2 » on trouvait des cas sous la forme « va,lue1value2 »). Face à ce constat qui aurait entrainé la prise en compte de valeurs aberrantes, nous avons finalement décidé de ne pas prendre en compte la colonne « *MAIN_ACTIVITY » dans l'élaboration du modèle.*

En ce qui concerne les colonnes dupliquées nous avons tout simplement décidé de supprimer les doublons et de ne garder que la première occurrence.

## Analyse de l'indépendance des variables

En prenant pour acquis que les variables numériques sont indépendantes de toutes les autres, nous avons réalisé un **test de cramer** sur l'ensemble des variables catégorielles (Agresti, 2017). Ce test permet de mesurer l'intensité d'une relation entre deux variables

Ainsi pour deux variables v1 et v2, si on note N la taille de l'échantillon, min_dll le degré de liberté du tableau et X^2 le score de chi2 entre v1 et v2 sous hypothèse d'indépendance, on obtient la formule suivante : v=√(x^2/(N*mindLL ))

En sociologie, v1 et v2 sont faiblement corrélées si Vc <0.2, moyennement corrélées si 0.2<Vc<0.3 et sinon, v1 et v2 sont fortement corrélées. Ainsi, le test de cramer nous permet non seulement de savoir si deux grandeurs sont liées, mais aussi de mesure l'intensité de cette relation.

La figure ci-dessous donne la matrice du test de cramer appliquée aux variables d'entrées catégorielles :

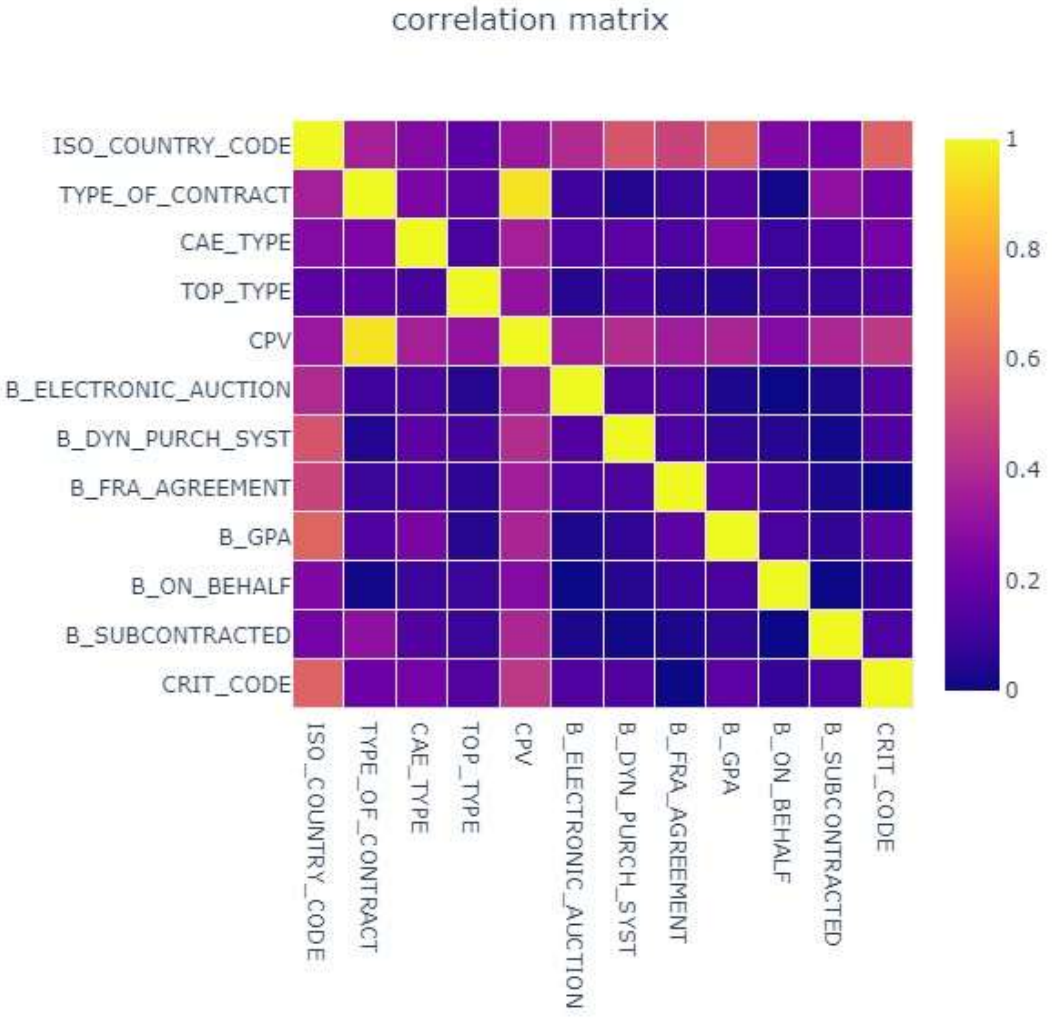


Figure 1 – Matrice de cramer des variables catégorielles

L'interprétation de cette matrice nous montre que les variables « CPV » et « TYPE_OF_CONTRACT » sont fortement corrélées (score de cramer de 0.95). Il est donc préférable de ne prendre en compte qu'une des deux. Dans une optique de réduction de la complexité du modèle, nous avons décidé de garder la colonne « TYPE_OF_CONTRACT » car elle ne possède que 9 valeurs uniques, et de mettre à l'écart la colonne « CPV » qui en possède elle 7690.

D'autre part, nous observons également que la variable « ISO_COUNRTRY_CODE » (représentant le pays dans lequel l'appel d'offre a été émis) est fortement corrélée à l'ensemble des variables catégorielles (tous les scores de la ligne correspondante sont supérieurs à 0.3). Cela traduit selon nous le fait que la règlementation propre à chaque pays influence la valeur de chaque variable. De ce fait nous avons décidé de ne pas prendre en compte le code iso du pays, car l'information apportée par cette variable se retrouve implicitement définie dans toutes les autres variables.

## Traitement des valeurs manquantes

Le jeu de données d'entrainement laissait apparaître des valeurs manquantes aussi bien dans les variables en entrées que dans les variables cibles.

Ainsi pour les variables cibles, nous avons fait le choix de supprimer toutes les lignes du jeu de données où une variable cible est non renseignée. Pour chaque variable à prédire, on a donc généré un jeu de données en s'assurant de ne garder que les valeurs non nulles. Le nouveau jeu de données pour prédire le nombre de candidatures à un appel d'offre représente donc 85% du fichier initial (environ 1,9 Millions de lignes) tandis que le second pour prédire la valeur du marché est identique au jeu de donnée initial car aucune valeur nulle n'a été recensé pour cette variable cible.

Pour les valeurs manquantes dans les variables en entrée, nous avons décidé de remplacer les valeurs manquantes. Le tableau ci-dessous récapitule les choix :

| Colonne | Valeur remplaçant les valeurs nulles | Explication |
|---|---|---|
| LOTS_NUMBER | Valeur moyenne de la colonne | Variable numé |
| CRIT_PRICE_WEIGHT | Valeur moyenne de la colonne | Variable numé |
| NUMBER_AWARDS | Valeur moyenne de la colonne | Variable numé |
| TYPE_OF_CONTRACT | undefined | |
| CAE_TYPE | undefined | |
| TOP_TYPE | undefined | |
| B_ELECTRONIC_AUCTION | OPE | OPE est le mo |
| B_DYN_PURCH_SYST | N | N est la valeur |
| B_FRA_AGREEMENT | undefined | |
| B_GPA | undefined | |
| B_ON_BEHALF | Undefined | |
| B_SUBCONTRACTED | undefined | |
| CRIT_CODE | Undefined | |

## Encodage des variables catégorielles : *One hot encoding*

Les algorithmes d'apprentissages statistiques ne traitent en entrée que des valeurs numériques. Or notre jeu de donnée étant en grande majorité constitué de variable catégorielle, il était donc nécessaire pour nous de procéder à une transformation de ces variables afin de pouvoir exploiter nos données. Nous avons choisi pour cela la méthode du *one hot encoding* car elle permet un encodage simple des valeurs catégorielles sans biaiser l'information de la donnée par une relation d'ordinalité inexistante.

Le ***one hot encoding*** prend une variable catégorielle X, et génère une nouvelle variable booléenne (0 ou 1) pour chaque valeur de la catégorie (Kedar Podtar, 2017) (sous la forme X_value). Cette technique a cependant l'inconvénient d'augmenter le nombre de variables, ce qui peut significativement augmenter la complexité du modèle dans le cas où on aurait des variables pouvant prendre un nombre conséquent de valeurs. De plus, elle rend plus complexe l'interprétabilité des résultats, car l'impact d'incidence sur la valeur à prédire n'est plus lié aux variables mais plutôt à des valeurs précises des variables.

Le one hot encoding s'implémente assez facilement en utilisant la fonction « OneHotEncoder » de la bibliothèque python sklearn.

```
From sklearn.preprocessing import OneHotEncoder
# On importe la fonction
Dataset = pd.read_excel(path)
#path est le chemin relatif du fichier contenant le jeu de donnée
categorical_features = [col1, col2, coln]
#où col1, col2 et coln sont les noms des colonnes du dataset que l'on souhaite encoder
encoder = OneHotEncoder(handle_unknown='ignore', categories=[dataset[col].unique() for col in categorical_features])
#on définit l'encodeur en fonction des paramètres souhaités
Dataset_encoded = encoder.fit(dataset)
#dataset_encoded est le nouveau Data frame avec les colonnes issues du OneHotEncoder
```

Le vecteur obtenu via le *one hot encoding* des variables catégorielles a ensuite été concaténé avec les valeurs normalisées (centrées et réduites sur les caractéristiques de leurs distributions respectives) des variables numériques présentes initialement dans le problème afin de constituer la base d'apprentissage de notre modèle.

# Résultats de prédictions

Une fois terminé le process de préparation des données, nous avons extrait un jeu de données différent pour chaque variable cible, avant de les splitter en jeu d'entrainement et jeu de test dans une proportion de 90/10. Nous avons ensuite entraîné 3 modèles sur chaque jeu d'entraînement avant de mesurer le **RMSE**.

Vu le large spectre des valeurs prises par la variable des montants de marchés (compris entre 10^2 et 10^8), le modèle s'est entraîné dans ce cas à prédire la valeur logarithmique de la colonne cible. Les scores obtenus sont consignés dans le tableau ci-dessous :

| Modèle | Train set nombre d'offres | Test set nombre d'offres | Train set du log montant du marché | Test set du log montant du marché |
|---|---|---|---|---|
| Random Forest | 19.814 | 22.241 | 1.57 | 1.64 |

| | | | | |
|---|---|---|---|---|
| Decision Tree | 19.802 | 19.25 | 1.56 | 1.63 |
| Artificial neural network | 22.57 | 22.241 | 1.84 | 1.86 |

Le RMSE ou erreur quadratique moyenne, représente la racine de la moyenne arithmétique des carrés des écarts entre prévisions du modèle et observations. Donc la précision du modèle est inversement proportionnelle au RMSE. Au vu des résultats du tableau ci-dessus, nous avons donc décidé de choisir le modèle obtenu par l'algorithme Decision Tree qui a obtenu le score le score plus bas sur les jeux de tests.

Nous avons réalisé les courbes d'apprentissage des deux modèles d'arbre de décision sélectionnées à base des scores du tableau :


Figure 2 – *Courbes d'apprentissage des modèles entraînés*

Il en ressort que l'arbre de décision entraîné sur les 2 prédiction généralise bien sur les données non vues. On observe un biais faible cars la valeur vers laquelle converge l'erreur du jeu d'entraînement est faible. Mais également une variance faible car c'est vers cette même valeur que converge l'erreur du jeu de test.

# Discussion

Bien que l'analyse que nous avons menée présente des résultats avec un bon niveau de précision, il est possible d'envisager quelque perspective d'évolution en vue d'améliorer la qualité du modèle

Tout d'abord, en ce qui concerne la variable « MAIN_ACTIVITY » que nous avons dû écarter parce qu'elle était inexploitable, la pertinence de cette donnée du fait de sa signification business nous fait penser que son exploitation pourrait améliorer les performances de nos modèles. Ainsi, si nous avions eu plus de temps à disposition, nous aurions traité la donnée en procédant en 2 étapes :

- Extraire dans un tableau la dénomination complète de toutes les valeurs de la colonne. (Cette tâche est plutôt chronophage car elle nécessite un travail manuel sur le jeu de données)
- Remplacer les valeurs incomplètes ou mal tronqués par leurs valeurs correctes, en s'appuyant sur les algorithmes de calcul de distance à l'instar de **Levenshtein** ou **fuzzywuzzy** qui permettent d'établir la proximité entre les expressions. Par exemple, l'expression "Housing and co" est plus proche par calcul de distance de "Housing

and community amenities" que l'expression "trolleybus or bus services". Ainsi après comparaison de l'ensemble des distances, la valeur "Housing and community amenities" sera assigné à toute occurrence de "Housing and co"

D'autre part l'application des techniques hyperparamétrisations auraient pu significativement booster les performances du modèle.

# Conclusion

En définitive nous avons mis en évidence, la possibilité de prédire avec un bon niveau de précision des indicateurs clés concernant les marchés publics, telles que le nombres de candidatures et le montant final du marché. La réussite de notre démarche réside principalement, comme dans la plupart des projets de data science, dans le soin apporté au processus de nettoyage des données, qui est primordiale avant toute prédiction de machine learning.

## BIBLIOGRAPHIE

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development*, *7*(3), 1247-1250.

Finlay, B., & Agresti, A. (1986). *Statistical methods for the social sciences*. Dellen.

Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, No. 8, pp. 707-710).

Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, *175*(4), 7-9.