

CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CENTRE REGIONAL DE PARIS

PROJET UASB03

présenté en vue d'obtenir
le certificat de spécialisation
d'analyste de données massives

par

Sébastien GARDOLL

DONORS CHOOSE
étude d'une plate-forme de financement
participatif

RESPONSABLE : Ndeye NIANG KEITA

ENSEIGNANTS : Michel CRUCIANU
Pierre CUBAUD
Marin FERECATU
Raphaël FOURNIER-S'NIEHOTTA
Ndeye NIANG KEITA
Nicolas THOME

Remerciements

Au terme de la certification d'analyste de données massives, j'aimerais sincèrement remercier toutes les personnes impliquées dans cette formation pour la qualité de leur enseignement ainsi que leur disponibilité et leur aide.

Je souhaite plus particulièrement remercier ma tutrice, Mme Ndeye Niang Keita.

Mes remerciements vont ensuite à Anastase Charantonis qui m'a offert son aide et de précieux conseils pour ce projet de certification.

Je tiens également à remercier toutes les personnes qui m'ont apporté leur soutien pendant mes intenses séances de révision, de codage, de rédaction tout au long de ces deux dernières années :

Martina Caneri et les autres camarades de STA211,

mes collègues de l'IPSL : Atef Bennasser et Guillaume Eberhardt,

mes amis montagnards de Toulouse, Alexandre Chetail et Yohann Chastre, promis je reprends l'entraînement (et pas que les modèles),

mes amis Laila et Nora Yahy, Antoine Moukarzel,

mes anciens collègues de Telecom-ParisTech,

mes amis danseurs de forró, et encore bien d'autres personnes.

Un remerciement spécial pour mon comité de relecture qui a courageusement chassé les bugs de mon rapport : Yann Sivry, Rémi Freydier ainsi qu'Éléonore Lauriat. Merci à toi également pour tes petites attentions, tes encouragements et à ta présence à mes côtés.

Pour finir, je pense vivement à ma famille qui m'a toujours soutenu lors de mes reprises d'études.

Encore merci.

Paris, le 28 septembre 2018

Table des matières

1	Introduction	1
1.1	Donors Choose	1
1.1.1	Présentation	1
1.1.2	Études précédentes	1
1.2	Problématiques	2
1.3	Méthode de développement	2
2	Données et infrastructure	4
2.1	Datasets	4
2.2	Variables	5
2.3	Infrastructure	5
2.3.1	Architecture	6
2.3.2	Environnement de programmation	7
2.3.3	Scalabilité	7
2.4	Importation en base	8
3	Statistiques	9
3.1	Variables calculées	9
3.2	Analyses unidimensionnelles	10
3.2.1	Table projects	10
3.2.2	Table teachers	10
3.2.3	Tables donations et donateurs	10
3.2.4	Table schools et ressources	10
3.3	Analyses bidimensionnelles	11
3.3.1	Jointures	11
3.3.2	Corrélations	11
3.4	Variables sélectionnées	12
4	Vectorisation des textes	13
4.1	Descripteurs de texte	13
4.2	Prétraitements	13
4.2.1	Racinisation	14
4.2.2	Lemmatisation	15

4.3	Vectorisation	16
4.3.1	Techniques de vectorisation	16
4.3.2	Résultats	17
5	Modélisation	19
5.1	Datasets	19
5.2	AFMD	20
5.3	Arbre de décision	21
5.4	Random Forest	21
5.5	Gradient Boosting	24
5.6	Récapitulatif	26
6	Conclusion	27
A	Variables	30
A.1	Exemple de page d'un projet	30
A.2	Description des variables	33
A.3	Résumés des variables quantitatives	36
A.4	Résumés des variables qualitatives	39
A.5	Répartition des modalités des variables qualitatives	41
A.6	Distribution des variables quantitatives	55
B	Corrélations	61
B.1	Test de Spearman	61
B.2	Tests du χ^2	61
B.3	Régressions Logistiques	63
C	Configurations	65
C.1	Spécifications matérielles	65
C.2	Spécifications logicielles	65
D	Modélisation	66
D.1	AFDM	66
D.2	Arbre de décisions	66
	Bibliographie	69
	Table des figures	73
	Liste des tableaux	75

1

Introduction

Dans le cadre de l'UA projet de la certification de spécialisation d'analyste de données massives du CNAM, j'ai choisi d'étudier une plate-forme spécialisée dans le financement participatif de projets d'établissements scolaires aux États Unis d'Amérique (EU). Ce chapitre présente la plate-forme Donors Choose, les problématiques traitées dans cette étude et énonce ses chapitres.

1.1 Donors Choose

1.1.1 Présentation

Donors Choose [10] est une plate-forme web qui organise le financement participatif (crowdfunding) de projets proposés par des élèves et/ou des professeurs d'écoles des EU, à l'image d'autres plate-formes comme Kickstarter [18] ou Ulule [37]. Donors Choose est une organisation à but non lucratif basée au EU et fondée en 2000 par Charles Best. Ce dernier, un ancien professeur du secteur public à New York - Bronx, souhaitait mettre en relation directement les professeurs des écoles avec des donateurs.

Le principe de Donors Choose est simple : le porteur de projet propose un projet en lien avec l'éducation (par exemple l'achat d'iPad pour faciliter l'apprentissage numérique) en le décrivant sur une page web de la plate-forme (photos, coûts, motivations, etc.). Un exemple est donné en annexe A.1. Après son acceptation, Donors Choose rend public le projet, permettant aux internautes, de choisir de le financer, principalement sous forme de versement d'argent (il existe d'autres formes de participation comme le leg, la cession de propriété, etc.). Bien entendu, le projet est à durée limitée et si la somme des dons n'atteint pas le coût de son financement au terme de celle-ci, le projet expire, ne reçoit aucun financement et les dons sont annulés ou réalloués à d'autres projets (à la discrétion des donateurs). Les projets sont décrits par un ensemble de variables, détaillées à la section 2.2.

Donors Choose est une plate-forme qui encourage l'analyse de ses données (CC BY-NC 3.0). Son dataset, relativement riche (localisation des donateurs, des écoles, description textuelles, etc.), est assez intéressant et permet l'application d'un grand nombre de techniques d'apprentissage automatique. Il a déjà fait l'objet d'études qui sont présentées à la section suivante.

1.1.2 Études précédentes

En premier lieu, Donors Choose centralise des questions d'analyses uni, bi, voir multidimensionnelles, sur son site web [6]. Ce site, par exemple, montre la répartition géographique des donateurs [9] ou étudie la relation entre la technologie et l'éducation [11]. Donors Choose peut mettre en

avant le projet, notamment lorsque celui-ci arrive à échéance sans être totalement financé. A cette fin, en 2016 une étude [26] menée par SAS Global Forum a proposé un système de recommandation basé sur un réseau de neurones. Cette étude s'appuyant sur les outils SAS notamment pour l'analyse de texte (HPTMINE) parvient à un modèle dont le score AUC (Area Under Curve) est de 0,74.

Il existe également un mémoire de master [19] qui décrit, en plus d'une synthèse sur l'apprentissage automatique, un modèle Random Forest (RF) équilibré prédisant l'intérêt des projets, avec un dataset sous échantillonné et enrichi (downsampling) ayant un score AUC de 0,615.

Enfin, Kaggle a organisé deux compétitions (mai et juin 2018). La première [8] avait pour but de prédire si un projet soumis à Donors Choose aller être accepté ou non. Le meilleur modèle [30] a obtenu un score AUC de 0,82 avec un empilement de modèles (stacking). La deuxième compétition [5] avait comme objectif d'aider Donors Choose à relancer les donateurs (la plupart ne donnent qu'une fois). Les compétiteurs ont soumis des solutions assez diverses comme des résumés statistiques des données, des systèmes de recommandation, etc.

1.2 Problématiques

Mon projet se concentre principalement sur l'explication et la prédiction du succès de financement d'un projet sur la plate-forme. Il propose également une description statistiques de la plate-forme incluant la relation entre les donateurs et les projets. De ces objectifs découlent les problématiques suivantes :

- la description du dataset ainsi que les corrélations entre des paires de variables
- la représentation vectorielle des textes
- la gestion de la mémoire et de la capacité de calcul
- la modélisation du financement des projets

1.3 Méthode de développement

Cette étude suit un processus classique d'exploration et de modélisation, illustré à la figure 1.1 et est organisée en quatre chapitres et une conclusion. Le chapitre *données et infrastructure* présente le dataset, les variables calculées et l'environnement de calcul utilisé pour réaliser cette étude. *Statistiques* apporte un éclairage sur la plate-forme de financement et ses données par le biais d'études uni et bidimensionnelle. La nature, la distribution et les corrélations des variables sont étudiées afin de sélectionner au mieux les méthodes d'apprentissage automatique en lien avec la modélisation du financement des projets. Le chapitre *Vectorisation des textes* couvre les opérations de modélisation des textes accompagnant les projets. Enfin, le chapitre *Modélisation* présente la mise en œuvre des algorithmes sélectionnés et leurs résultats.

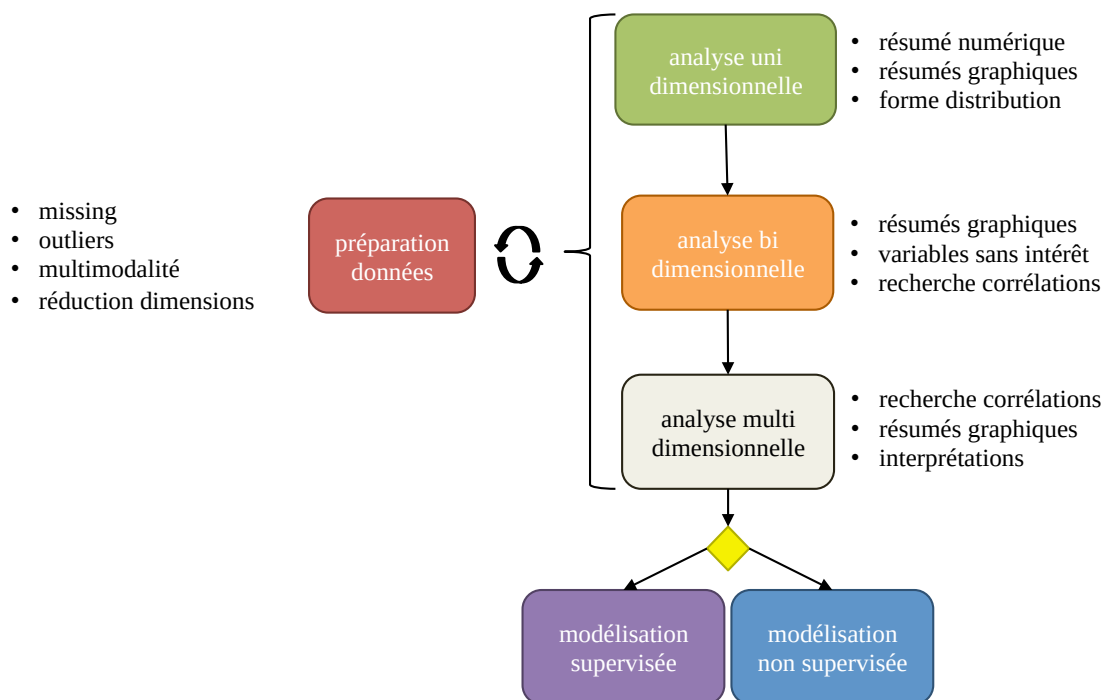


Figure 1.1 : Processus d'exploration et de modélisation

2

Données et infrastructure

Cette étude s'appuie sur deux datasets. L'un provenant du site d'analyses de données de Donors Choose et le deuxième provenant de Kaggle. Ce chapitre présente ces datasets, décrit les variables qui les composent et explique pourquoi utiliser les deux datasets. L'environnement de calcul est traité et la question de la scalabilité est également discutée. Enfin le chapitre se termine sur les aspects d'importation des données en base.

2.1 Datasets

Il existe trois datasets différents. L'un est proposé par Donors Choose via son site de traitements de données [4] et deux autres par Kaggle lors de sa première [8] et sa deuxième compétition [5]. Un examen rapide permet d'écartier le dataset de la première compétition Kaggle car sa richesse, en terme de variables, est loin de celle des deux autres dataset. Le deuxième dataset Kaggle est relativement riche mais il n'égale pas la richesse de celui de Donors Choose et ne couvre qu'une plus courte époque.

De prime abord, l'étude devait porter uniquement sur le dataset proposé par Donors Choose, dénommé ici « dataset Donors Choose ». Or l'analyse unidimensionnelle de ce dataset a montré des problèmes d'incohérences (jointure de tables) et des informations intéressantes, comme le type de ressources demandées par les projets, de relativement à massivement incomplètes.

Les datasets de Kaggle sont en général bien pré-traités, et celui de la deuxième compétition de Donors Choose sur Kaggle l'est également comme le montre les différents graphiques de dénombrement sur son site [5] : les variables présentent peu de valeurs manquantes et aucun problème de cohérence n'a été rapporté.

Les datasets de Kaggle et Donors Choose sont distribués sous la forme de fichiers CSV qui reflètent une organisation en base relationnelle. En effet, chacun des fichiers comporte une clef primaire et des clefs étrangères provenant d'autres fichiers, afin d'éviter une redondance de l'information.

Comme les datasets de Kaggle sont issus du dataset de Donors Choose, les clefs ou identifiants sont exactement les mêmes. Ainsi, il a été possible d'enrichir le dataset de la deuxième compétition de Kaggle, dénommé « dataset Kaggle » ou dataset lorsqu'il n'y a pas besoin de lever d'ambiguïté, avec celui du dataset Donors Choose (quand celui-ci couvre la même période temporelle). La technique utilisée pour enrichir le dataset Kaggle est une jointure à gauche sur les identifiants (dataframe Spark). Le dataset Kaggle est composé de six fichiers résumés à la table 2.1. Le dataset Donors Choose est composé de sept fichiers dont le poids total non compressé est de 8 Go.

nom de table	nb lignes	nb var	nb var enrichi	nom de fichier	Mo
projects	1 110 015	18	41	Projects.csv	2600.0
donations	4 687 844	7	8	Donations.csv	611.4
teachers	402 900	3	11	Teachers.csv	19.6
donors	2 122 640	5	5	Donors.csv	124.0
schools	72 993	9	23	Schools.csv	9.7
resources	7 210 448	5	6	Resources.csv	819.3

Table 2.1 : Description des tables du dataset Kaggle

2.2 Variables

Cette section décrit uniquement les variables du dataset Kaggle. La description complète de celles du dataset Donors Choose est disponible sur son site [7]. Le nombre des variables par table du dataset Kaggle est indiqué à la colonne « nb var » à la figure 2.1. La colonne « nb var enrichi » indique le nombre de variables après enrichissement avec celles du dataset Donors Choose et avec celles calculées pour cette étude (voir la section 3.1). Les variables sont un mélange de variables quantitatives, nominales, ordinales, de champs de codes (codes postaux), de dates, ainsi que de quatre champs texte. Elles sont décrites et résumées en annexe A.2 (les variables en orange proviennent du dataset Donors Choose et les variables en bleu sont les variables calculées) et leurs analyses sont données au chapitre 3. Cependant voici la liste des variables saillantes du dataset Kaggle :

- status (projects) : l'état du projet (Fully funded, Expired ou Live) à la date de l'extraction du dataset.
- project_cost (projects) : le coût total du projet soumis à la plate-forme.
- students_reached (projects) : le nombre d'élèves concernés par le projet.
- grade_level (projects) : le niveau de la classe à l'origine du projet.
- subject_cat (projects) : la catégorie du projet (literature&langage, etc.).
- resource_cat (projects) : le type de ressources demandées (livres, etc.).
- title, statement, short_description et essay (projects) : les champs texte accompagnant la description du projet.
- school_type (schools) : le type d'implantation de l'école (rurale, etc.).
- school_city, school_county, school_state (school) : la localisation de l'école.
- poverty_level (schools) : le niveau de pauvreté de l'école.
- free_lunch (schools) : le pourcentage de repas subventionnés d'une école.
- donor_city, donor_state (donors) : la localisation du donateur.
- donation_amount (donations) : le montant du don exprimé en dollars.
- is_teacher (donations) : indique si le don provient d'un professeur.
- teacher_prefix (teachers) : le titre du professeur des écoles (Mrs, Mr, etc.).

Les variables calculées en lien avec les textes des projets sont l'objet du chapitre 4 et les autres sont abordées à la section 3.1.

2.3 Infrastructure

Un ordinateur de bureau tient les deux datasets en mémoire mais sa capacité de calcul est insuffisante pour obtenir des temps de calcul correct (l'entraînement d'un modèle gradient boosting sur le dataset Kaggle prend environ deux heures avec le cluster utilisé dans cette étude). Afin de ne

pas restreindre mon étude en sous échantillonnant mais également afin de l’ancrer dans les problématiques big data, j’ai constitué un cluster Spark ad hoc (configuration matérielle en annexe C.1).

2.3.1 Architecture

Le cluster est des plus simples (figure 2.1) : trois machines de bureau aux capacités similaires, communiquant en réseau et montant un espace de stockage commun par NFS. Ce dernier sert à l’installation des exécutables (Spark, MongoDB, etc.), à la gestion de packages Spark (bibliothèques au format jar) communs aux nœuds du cluster et à la persistance de fichiers de résultats et des dataframes Spark temporaires au format parquet. La gestion des données est confiée au moteur de bases NoSQL MongoDB. L’orchestrateur de tâches du cluster est celui fournit avec la distribution de Spark. Les aspects de scalabilité sont abordées dans la sous section suivante (2.3.3).

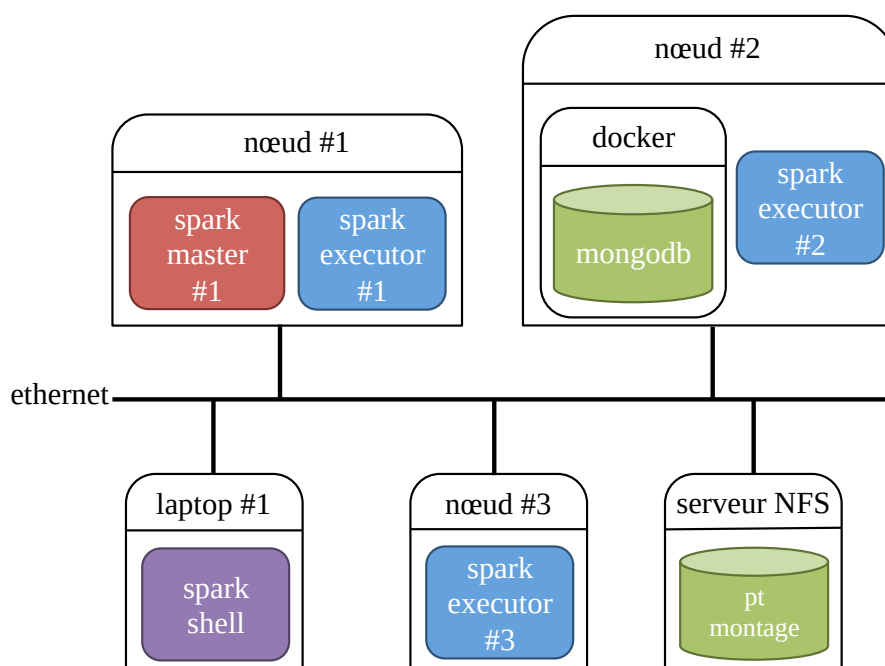


Figure 2.1 : Architecture du cluster

Le choix d’une plate-forme de calcul Spark est bien adapté concernant le dataset à étudier. En effet, la distribution des calculs aux nœuds du cluster et la persistance des données en mémoire sont les clefs de la performance pour des opérations de jointures, d’agrégation et de tris, effectuées lors de l’exploration des données. De plus, Spark apporte une API supportant le concept de dataframe depuis sa version 1.6 et offre un sous ensemble d’opérations SQL (groupBy, where/filter, etc.) facilitant grandement la fouille de données par rapport à l’ancienne API qui expose uniquement le concept de RDD et des opérations bas niveau (map, reduce, etc.). Il comporte également une bibliothèque relativement riche en algorithmes de machine learning qui est exploitée dans cette étude. De plus, l’implémentation du concept de pipeline de tâches dans Spark automatise et optimise les transformations des données.

MongoDB répond au besoin de distribuer et persister efficacement les données dans un environnement concurrentiel. En effet, grâce au connecteur mongo-spark, les workers Spark lisent et écrivent en même temps les données. Mais surtout, MongoDB est un outil permettant des requêtes souples (syntaxe style Javascript), répondant relativement vite et permettant l’indexation de tous les champs, même des mots dans un tableau de chaînes de caractères. Cette fonctionnalité s’est avérée très pratique pour la mise au point de la vectorisation des textes (voir chapitre 4). Mon-

goDB est bien sûr adapté au type de dataset de cette étude dont l'organisation est calquée sur celle des bases de données relationnelles (voir section 2.1). Il permet également le stockage de routines, utile pour la vérification des types des champs et des valeurs manquantes. De plus, son modèle de scalabilité est relativement simple et passe à l'échelle sans problème pour un plus grand volume de données. L'instance de MongoDB est exécutée au sein d'un container Docker afin de simplifier son éventuelle migration sur un autre nœud du cluster. Enfin, l'orchestrateur de tâches de Spark a été suffisant car j'étais le seul à utiliser le cluster.

2.3.2 Environnement de programmation

Scala est le langage d'implémentation de cette étude. Langage référentiel de Spark, l'écart de fonctionnalité entre l'API de Spark écrite en Scala et celles en R (SparkR) et Python (PySpark) se réduit de versions en versions de Spark.

La version de Spark utilisée dans cette étude (spécifications logicielles en section C.2) supporte les User Defined Function (UDF) sous PySpark. Cependant, selon [1] et [13], l'implémentation des UDF et des pipelines dans un langage différent de Scala (et Java), n'est pas optimum parce que cela induit des opérations de sérialisation/désérialisation coûteuses en temps de calcul, entre les machines virtuelles Java et Python. Il est bien sûr possible d'implémenter des UDF et des pipelines sous Scala (ou Java), et de les utiliser sous PySpark sous forme de bibliothèques jar. Mais dans un souci de simplicité et de centralisation du code de l'étude, j'ai préféré implémenter uniquement l'étude en Scala, mis à part le code de génération des graphiques sous Python avec Seaborn. A noter que l'intégration de Panda, la bibliothèque de dataframe de Python, est en cours dans PySpark.

2.3.3 Scalabilité

Si l'extension des ressources matérielles (CPU et RAM) s'effectue simplement par ajouts de nouveaux nœuds, la gestion de l'infrastructure du cluster est inexistante et l'orchestrateur de tâches de Spark, est insuffisant. En effet, dans un contexte multi-utilisateurs et d'exécution de tâches concurrentes, cet ordonnanceur ne propose pas de mécanisme de partage de ressources matérielles. Par défaut, les tâches essayent de prendre toutes les ressources des nœuds sur lesquels elles sont placées, se concurrençant directement. L'ordonnanceur Spark offre, tout au plus, une stratégie d'ordonnement des tâches en FIFO. Un orchestrateur de tâches comme Yarn est nécessaire et apporte la flexibilité nécessaire afin de partager les ressources entre les différentes tâches et un bien meilleur système de failover. HDFS complète le cluster en apportant un système de stockage conscient de la localisation des données.

Cependant, le cluster reste dédié à l'exécution de tâches Spark. Pour diverses raisons comme la rentabilité du cluster ou la réalisation de calculs selon d'autres modèles de calcul que celui de Spark (ex : OpenMPI), il est généralement nécessaire d'ouvrir le cluster à d'autres technologies de calcul distribué. Mesos contribue à ce partage en encapsulant des tâches Spark, des instances de MongoDB et, par exemple, des applications OpenMPI, par des exécutants Mesos comme l'illustre de façon simplifiée la figure 2.2.

Enfin un système de déploiement et un autre de redémarrage de logiciels finissent de répondre aux besoins d'un cluster scalable. En effet, il faut être capable de déployer les composants de Spark, Mesos et MongoDB sur tous les nœuds du cluster, de façon automatique et conditionnelle et aussi de les redémarrer en cas de panne ou d'absence de réponse de leur part (heart beat). On peut citer Ansible, un gestionnaire de configuration de nœuds de cluster et Marathon, un système de déploiement et d'orchestration de containers Docker pour Mesos.

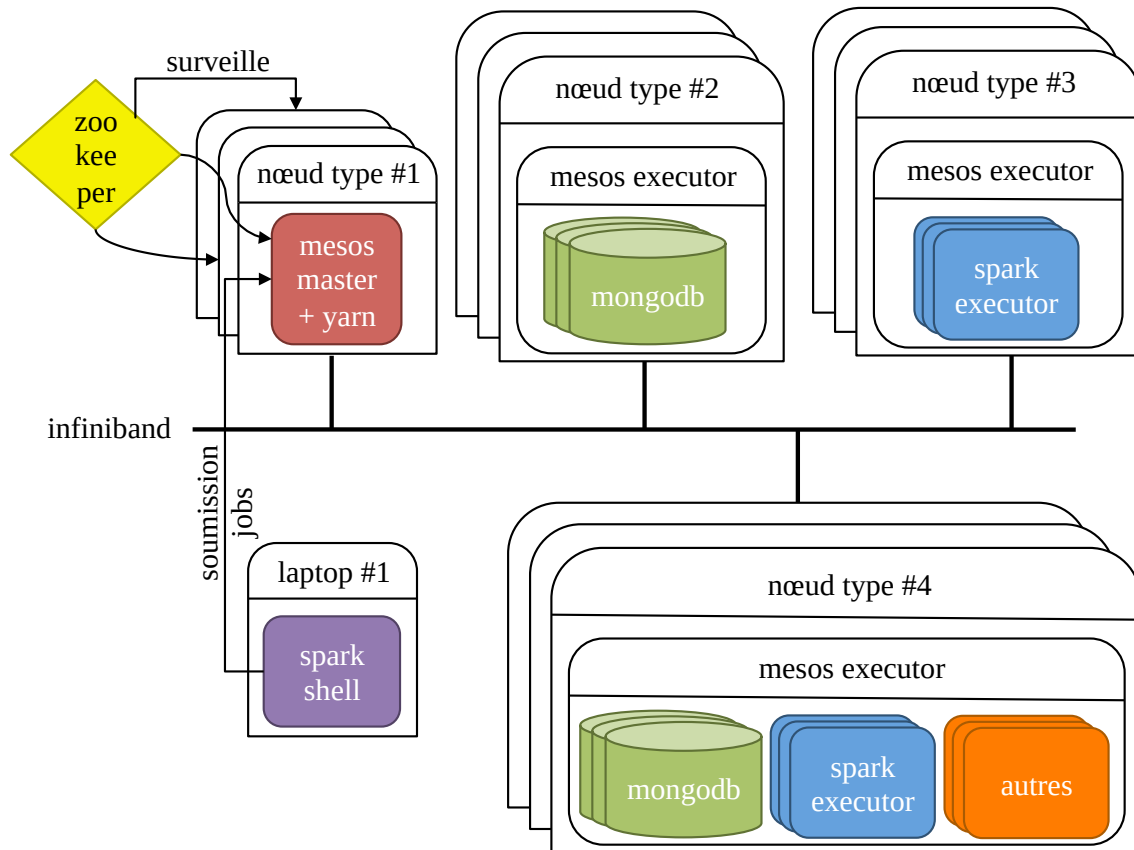


Figure 2.2 : Architecture scalable

2.4 Importation en base

Mongodb offre un outil simple et complet (mongoimport) pour importer des données sous forme de fichiers CSV. Il permet, notamment, d'assurer l'unicité des identifiants des enregistrements et donc de vérifier la présence de doublons dans les fichiers. Il permet également de définir le schéma des données : le nom des métadonnées et leur type [23].

L'inférence de type des données de Mongodb s'est quelque fois trompée au sujet des données de cette étude (ex : code postaux considérés comme des réels). Figer le type des données assure donc de la bonne concordance entre les données et leur encodage dans Mongodb puis dans Spark. Ainsi, les variables quantitatives sont définies à l'aide du type BSON le plus adapté (integer ou double). Les variables qualitatives, même celles exprimées numériquement, sont encodées en type string afin d'interdire tous calculs arithmétiques.

Concernant le dataset Donors Choose, un prétraitement est nécessaire afin de rendre les textes compatibles avec la norme CSV, en particulier supprimer les guillemets contenus dans les textes car ils sont interprétés comme des délimiteurs de textes par Mongodb. Le prétraitement est implémenté en bash et consiste en des appels de la commande sed. Enfin, les valeurs manquantes sont signalées par la valeur nulle.

3

Statistiques

Ce chapitre présente un résumé statistique non exhaustif du dataset Kaggle, amélioré par le dataset Donors Choose et augmenté par un ensemble de variables calculées. Cette étape de l'étude a pour objectifs de répondre à quelques questions sur la plate-forme de financement participatif et de justifier une sélection des variables pour la modélisation.

3.1 Variables calculées

En plus des variables empruntées au dataset Donors Choose, le dataset de cette étude a été enrichi par 27 variables calculées (en bleu dans l'annexe A.2). Parmi ces variables, on compte celles provenant de la transformation de variables difficiles à intégrer telles quelles, celles calculées à partir de jointures de tables dont la cardinalité est multiple et enfin celles extérieures au dataset. Cette section ne traite pas des descripteurs de texte qui font l'objet du chapitre 4.

Les variables issues des dates (`first_project`, `duration`, `posted_month`, `expired_month`, etc.) sont transformées en variables représentant le numéro du mois de la date afin d'évaluer une quelconque préférence lors de la création de projets ou du versement de dons.

Les variables caractérisant les ratios de succès de financement des projets au niveaux des états, des comtés et des villes, (`state_success_ratio`, etc.) et les compteurs de nombre de projets (`nb_county_projects`, etc.) suivant les mêmes niveaux, proviennent de la jointure entre les tables `projects` et `schools`. Ces variables modélisent en partie la cardinalité multiple entre `projects` et `schools`.

Les compteurs de donateurs suivant les états et les villes (`nb_city_donors`, etc.) ont été créés en remplacement des variables des tables `donations` et `donors` qui ne peuvent être prises en compte lors de la modélisation comme le démontre la section 3.3.1.

Les compteurs de projets (`project_count`, `funded_count`, etc.) et la variable `teacher_success_ratio`, sont des variables créées afin de caractériser le succès de financement des projets d'un professeur. Ces variables, utiles pour décrire le dataset, n'ont pas été utilisées lors de la modélisation en raison de la relation évidente avec la variable `status` à modéliser.

Enfin les variables `zip_density`, `zip_med_home_value` et `zip_med_income`, proviennent du recensement des localités des EU, et viennent caractériser les écoles soumettant un projet Donors Choose. Les données de ces variables ont été récupérées sur le site United States Zip Codes ([38]) à l'aide de scripts Python gérant la récupération des pages web, leur analyse et la gestion des données manquantes.

3.2 Analyses unidimensionnelles

Cette section liste les statistiques unidimensionnelles intéressantes. Les variables qualitatives sont présentées en annexe à la section A.4 et la distribution de leurs modalités à la section A.5.

De la même façon, les variables quantitatives sont présentées à la section A.3 et leur distribution à la section A.6. Spark ne proposant pas nativement de bibliothèque de visualisation de données, les données des histogrammes ont été construites sous Spark, en regroupant les valeurs des variables en 20 modalités (à l'aide de Bucketizer) puis en générant les graphiques avec LibreOffice. Afin de ne pas nuire à la lisibilité des graphiques, les valeurs inférieures au premier décile ou supérieures au 99^e centile (comme les box plot de Tukey) ne sont pas prises en compte. Les outliers ne sont donc pas représentés, cependant ils sont en partie détectables dans le résumé numérique des variables (min, max, quartiles). Le profil des variables quantitatives, les écarts types dépassant les moyennes et le test de Kolmogorov-Smirnov contre la loi Normale, montrent qu'aucune des variables quantitatives ne suit la loi Normale, même après transformations logarithmiques (décimale et naturelle).

3.2.1 Table projects

L'analyse de la table projects montre que 74 % des projets soumis sur la plate-forme sont financés. Leur coût moyen est de 740 \$ et concerne en moyenne 93 élèves provenant des niveaux équivalents, de la moyenne section de maternelle au CE1 (39 %) et du CE2 au CM2 (32,8 %). 98 % des projets sont dirigés par des professeurs (sinon par les élèves eux-mêmes) pour qui il s'agit d'un premier projet à 69 %. La durée moyenne d'un projet est de 116 jours et sont à presque 14 % déposés en septembre. 34 % des demandes sont étiquetées provisions et catégorisées à 22,5 % en littérature&langage et 15,5 % en mathématiques&science.

3.2.2 Table teachers

La table teachers nous apprend que 86 % des professeurs des écoles sont des femmes, quelque fois docteurs en sciences ou ayant un titre original (Mx.). La plupart d'entre eux n'ont soumis qu'un seul projet.

3.2.3 Tables donations et donateurs

Donations et donateurs montrent que le nombre de dons s'élève en moyenne à 2,31 dons par donateur. Cependant, sachant que le profil de distribution des dons est très étalé, il convient de prendre la médiane qui est égale à 1. En effet, le plus prodigue des donateurs (certainement une entreprise) a donné 18 035 fois. Le montant du don moyen est d'environ 60 \$. Si les dons sont versés à tout moment de l'année, 12 et 11 % des dons sont réalisés respectivement aux mois d'août et de septembre (ce qui est cohérent avec le pic de dépôt des projets). 10 % des donateurs sont des professeurs et leurs dons représentent 28,6 % du montant total.

Au classement des états, la Californie arrive en tête avec 16,4% en nombre de donateurs (et 13,8 % en montant des dons) suivit de l'Etat de New York avec 8,63% et celui du Texas, avec 6,12%.

3.2.4 Table schools et resources

Les écoles concernées par un projet Donors Choose, sont à 62,7 % urbaines, réparties à 31,5 % en banlieue et à 31,22 % en ville. 40,5 % des écoles sont qualifiées de très pauvres. En moyenne, 58 % des repas des élèves sont subventionnés. Le classement du nombre d'écoles (et indirectement des projets) par État est semblable à celui du nombre de donateurs : la Californie arrive en tête avec 11,5 %, le Texas suit avec 8,88 % et le troisième est New York, avec 5,23 %. Concernant

les ressources demandées par les projets, elles valent en moyenne 53,4 \$ à l'unité et proviennent essentiellement de Amazon Business (44,4 %) et Lakeshore Learning materials (15 %).

3.3 Analyses bidimensionnelles

De la même façon que la section précédente, cette section ne décrit qu'une sélection de relations. La méthodologie est classique, les corrélations sont évaluées par paires. Pour les paires de variables quantitatives, le test de Spearman est appliqué (aucune variable quantitative ne suit la loi Normale, voir annexe B.2). Pour les paires de variables qualitatives, le test du χ^2 et le v de Cramér (annexe B.2) sont évalués. Pour les paires mixtes, une régression logistique est entraînée avec comme métriques AUC et F1, respectivement pour les variables bimodales et multimodales (annexe B.3). Afin d'éviter toute perturbation induite par les outliers, les valeurs des variables quantitatives inférieures au premier décile et supérieures au 99^e décile, ne sont pas prises en compte.

3.3.1 Jointures

Les jointures entre les tables, nécessaires pour des paires de variables appartenant à des tables différentes, sont systématiquement vérifiées. Si les jointures des tables dont la cardinalité de la relation est 1 (teachers → schools, donations → donors, etc.) ne posent pas de problème : 99 % des lignes des tables sont bien mises en correspondance, les jointures des tables projects → donations-donors et projects → ressources dont la cardinalité de la relation est 1 à multiple, ne sont pas satisfaisantes (voir figure 3.1).

Tout d'abord, ces jointures peuvent être vérifiées à l'aide des variables project_cost et status. Ainsi la somme des ressources par projet n'atteint pas le coût du projet. De la même façon, la somme des dons par projet, pour 64 % d'entre eux, est inférieure au coût du projet alors que les projets sont qualifiés de financés.

Afin de contourner ces difficultés, j'ai décidé de ne pas joindre les tables ressources et donations avec la table projects. A la place, j'ai créé des variables résumant statistiquement une partie de l'information contenue dans la table donors (nb_state_donors, etc.) par rapport à ses variables de localisation (donor_state et donor_city) dont la jointure avec donations est satisfaisante. Comme les modalités des variables de localisation de la table donors sont exactement les mêmes que celles de la table schools, l'insertion de ces nouvelles variables dans le résultat de la jointure projects-schools est réalisée. Concernant la jointure projects → schools et projects → teachers dont la relation est multiple à 1, il n'existe pas de variable de vérification. J'ai considéré que le dataset était cohérent.

3.3.2 Corrélations

Les paires de variables quantitatives et qualitatives sont corrélées, mais à des niveaux infinitésimaux. Des corrélations étudiées (annexe B), seule project_cost et donation_amount sont faiblement corrélées avec un score de 0,13 au test de Spearman.

Le résultat des régressions logistiques pour les paires mixtes sont un peu plus intéressantes (annexe B.3). Ainsi donation_amount permet de classifier les modalités de is_teacher à 68 % et à 57 % pour celles de first_project. Free_lunch et zip_med_income classifient poverty_level respectivement à 86 % et à 55 %. project_cost classifie first_projects à 59 % et status à 67 %. Enfin, zip_density classifie status à 54 %.

Concernant la relation entre les donateurs et les écoles, les informations étant assez limitées, je n'ai pu étudier que leur proximité géographique. Le test du χ^2 entre donor_state et school_state montre une corrélation. Mais la valeur du v de Cramér est extrêmement petite (0,08). Cependant,

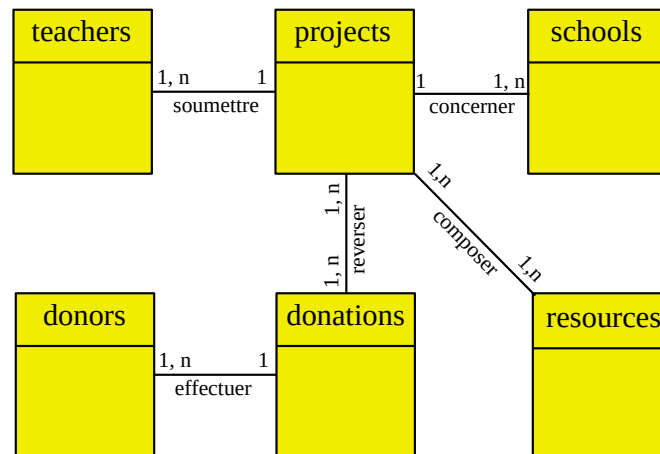


Figure 3.1 : Diagramme MCD du dataset Kaggle

le dénombrement des couples donateur - école (jointure indirecte entre donations et schools grâce aux variables de localisation de donateurs et de schools) montre que 24 % appartiennent à la même ville et 63 % au même État. En fait, le test du χ^2 mesure l'influence des modalités d'une variable sur la fréquence des modalités d'une autre variable (table de contingence). Le dénombrement liste simplement les coïncidences entre la localisation (État ou ville) et le lieu du donateur, et non cette influence. On peut donc conclure qu'une grande part des dons est locale (ville et État).

3.4 Variables sélectionnées

L'étude bidimensionnelle montre que, prises deux par deux, les variables ne sont que très faiblement corrélées. S'il existe une relation modélisable entre status et les autres variables, elle est de nature multidimensionnelle et non linéaire. De plus, l'analyse unidimensionnelle montre des distributions de variables quantitatives très étalées. Un prétraitement afin d'atténuer l'effet des outliers est donc nécessaire avant d'entraîner des modèles, sauf pour ceux dont les algorithmes sont basés sur les arbres de décision. Par conséquent, j'ai décidé de modéliser la variable status à l'aide du maximum de variables disponibles.

4

Vectorisation des textes

Sans doute la plus importante source d'information du dataset, les textes qui accompagnent chaque projet de la plate-forme Donors Choose, ont fait l'objet d'un traitement en particulier. Ce chapitre commence par lister les descripteurs de texte utilisés dans cette étude, puis décrit la transformation des textes en un ensemble de lemmes et racines qui alimentent un processus de vectorisation explicité à la fin du chapitre.

4.1 Descripteurs de texte

Pour chacun des textes (title, statement, short description et essay), il est possible de créer des descripteurs assez simples :

- le nombre de symboles
- le nombre de mots
- la moyenne de la taille des mots
- le nombre de mots uniques du texte
- la diversité lexicale comme le rapport entre le nombre de mots uniques d'un texte et le nombre de mots total du texte

Si le premier descripteur ne demande aucun prétraitement, les autres descripteurs en nécessitent un certain nombre. Ces traitements sont abordés à la section suivante.

4.2 Prétraitements

Calculer ces descripteurs de texte sous entend de découper le texte en mots, en supprimant la ponctuation et les nombres, puis de les normaliser et de les comptabiliser. Concernant le calcul de la diversité lexicale, la normalisation des mots a consisté à trouver leur racine lexicale, en partant du principe que la racine des mots en anglais est une bonne évaluation du point commun entre ces mots.

La lemmatisation, qui est l'autre alternative à la racinisation, calcule la forme canonique neutre des mots. Elle est utilisée lors de la vectorisation des textes de cette étude (section 4.3) car elle préserve mieux la sémantique des mots et donc la spécificité des thèmes abordés par les textes.

Afin de réaliser ces opérations d'analyse lexicale, j'ai recensé les bibliothèques NLP existantes sous Spark. En voici une liste non exhaustive :

- Spark NLP [20]
- Spacy [32]
- Stanford CoreNLP [12]
- OpenNLP [25]
- UIMA [36]
- Mallet [21]
- Gate [14]

Bien que très récente (moins de deux ans), Spark NLP est une technologie assez prometteuse. En effet, elle s'intègre parfaitement à Spark car elle est construite au dessus de sa bibliothèque de Machine Learning (Spark ML). Spark NLP tire parti de l'abstraction de la distribution de calcul offert par Spark et de son système de pipeline. Le gain de temps de calcul est appréciable.

Spacy est une bibliothèque bien établie et efficace dans le monde Python et peut donc être appelée sous PySpark mais au prix d'un certain coût de communication inter-processus comme la sérialisation/désérialisation d'information entre les machines virtuelles Python et Java (problème évoqué en sous section 2.3.2).

UIMA est une spécification d'interfaces pour l'analyse des langages naturels et n'est donc pas une implémentation d'une bibliothèque.

Les autres bibliothèques Stanford CoreNLP, OpenNLP, Gate et Mallet sont des bibliothèques écrites en Java dont l'efficacité est certaine mais ne sont pas spécialement intégrées à Spark.

J'ai donc choisi Spark NLP qui semble être promise à un distribution conjointe avec Spark. Spark NLP propose les fonctions de base des logiciels d'analyse de langage : lexeur, correcteur d'orthographe, racinisation et lemmatisation.

4.2.1 Racinisation

Afin de calculer la diversité lexicale des textes, j'ai choisi de raciniser les mots du texte. Cependant, la racinisation n'est qu'une étape parmi d'autres, amenant à son calcul. La figure 4.1 résume de façon simplifiée toutes les étapes du pipeline Spark aboutissant au calcul de la diversité lexicale sur l'ensemble des enregistrements de la table projects (environ un million de lignes). Chaque étape a fait l'objet d'un effort d'optimisation.

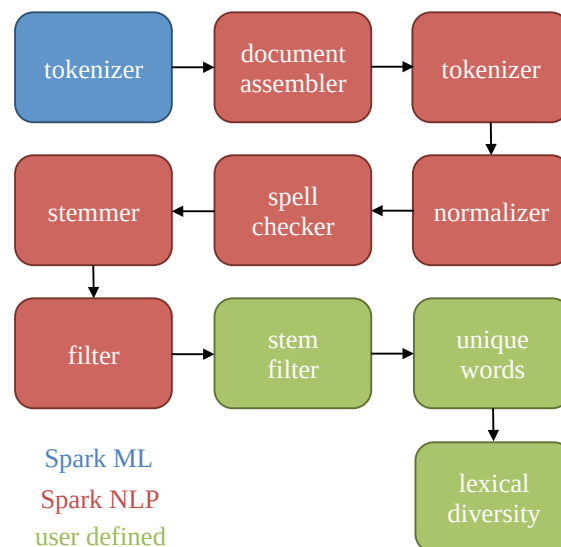


Figure 4.1 : Pipeline diversité lexicale

La première étape du pipeline consiste à découper le texte en mots en utilisant le tokenizer de la bibliothèque Spark ML. A l'aide d'une expression rationnelle, le tokenizer découpe les mots selon

les espaces et la ponctuation en préservant les deux symboles d’apostrophes, les nombres et toutes les lettres mêmes accentuées (de n’importe quels alphabets). Par contre, les mots composés fermés (reliés par un trait d’union) sont découpés en plusieurs mots.

L’étape « document assembler » consiste à préparer les mots précédents aux traitements de la bibliothèque Spark NLP. Ces traitements commencent par un découpage plus fin des mots à l’aide du tokenizer de Spark NLP, chargé de traiter les abus de contractions (ex : concaténation de mots à l’aide d’apostrophes) avec un ensemble d’expressions rationnelles couvrant le maximum de cas observés. Le normalizer convertit les lettres en minuscules puis le spell checker corrige l’orthographe des mots, à l’aide d’un modèle (Norvig) pré-entraîné par Spark NLP sur un corpus de Wikipedia, toujours dans le soucis de calculer le nombre de mots uniques le plus juste. Le stemmer prend la suite en produisant les racines. L’étape « finisher » consiste à enlever les annotations de Spark NLP. L’étape « stem filter » est une étape que j’ai spécialement implémentée pour supprimer les nombres et les mots comportant des chiffres, et résoudre les mots contractés en anglais (ex : won’t compte pour will et not) [35] et [39]. Enfin les étapes « unique words » et « lexical diversity », calculent, respectivement, les mots uniques et la diversité lexicale. Contrairement à la lemmatisation, les stop words ne sont pas retirés car je pense qu’ils font partie des mots uniques à comptabiliser.

4.2.2 Lemmatisation

La transformation des textes en un ensemble de lemmes est une des étapes nécessaires afin de vectoriser les textes. Cette technique a été préférée à la racinisation car contrairement à cette dernière, la lemmatisation conserve mieux la sémantique des mots. Or, préserver la particularité des textes est important afin d’expliquer le succès du financement des projets. La figure 4.2 illustre son implémentation sous forme de pipeline Spark.

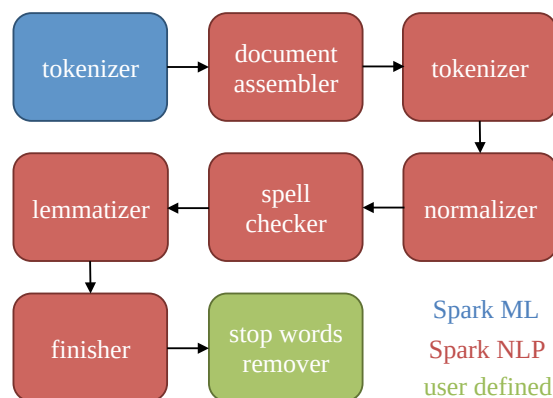


Figure 4.2 : Pipeline lemmatisation

Le début du pipeline de lemmatisation ressemble à celui du calcul de la diversité lexicale. Cependant, le premier tokenizer préserve les mots composés ouverts mais ne préserve pas les apostrophes, simplifiant ainsi la gestion de la contraction en anglais. Par contre, le découpage selon les apostrophes génère de petits mots d’une ou deux lettres (ex : ve et t provenant respectivement de we’ve et isn’t). Heureusement, les mots strictement de moins de trois lettres sont souvent peu porteurs d’information. Ils sont éliminés dans la dernière étape. Les mots composés ouverts (ex : « New York » doit être considéré comme un seul et même mot) sont épargnés du découpage afin de préserver leur sémantique, particulièrement pour les techniques de vectorisation type Bag Of Words (BOW). Par exemple, « car pool » signifie covoiturage mais pris séparément « car » et « pool » ont un sens totalement différent. La liste de mots ouverts a été fabriquée à partir de plusieurs sites web dont Wikipedia pour les noms de localités américaines. Le lemmatizer, un modèle pré-entraîné sur un corpus AntBNC par Spark NLP, produit les lemmes des textes qui sont filtrés à la fin du pipe-

line. Le filtrage consiste à supprimer les mots très faiblement porteurs d'information, diminuant alors le nombre de dimensions du vecteur représentant le corpus de textes. Ainsi les stop words n'ont pas été retenus de même que les lemmes de moins de trois lettres strictement. La liste des stop words est une concaténation de plusieurs listes disponibles sur le Web ([24], [34], [28] et [40]) et de celle de Spark ML (la classe StopWords). Elle contient environ 900 mots (dont leurs formes mal orthographiées).

4.3 Vectorisation

Si les descripteurs de texte décrits ci-dessus rendent compte des textes de façon globale et limitée, la vectorisation des textes permet de prendre en compte leur structure, leurs thèmes, leurs mots ou leurs locutions. Une fois vectorisés, ces textes peuvent ensuite alimenter tout un ensemble d'algorithmes de modélisation. Le choix de la technique de vectorisation est donc stratégique. Dans la mesure du possible, cette technique devrait préserver les particularités des textes (cités ci-dessus) et permettre l'opération inverse, afin de retrouver quelles particularités (mots, thèmes ou structure) ont une influence sur le financement des projets de la plate-forme. A cela s'ajoute la contrainte très forte de réaliser cette technique sur un cluster dont les ressources sont très limitées. Cette contrainte se traduit généralement par une limitation du nombre de dimensions des vecteurs produits par ces techniques. Spark ML propose les trois techniques suivantes : word2vec, Term Frequency - Inverse Document Frequency (TF-IDF) et Latent Dirichlet Allocation (LDA). La sous section suivante présente leurs propriétés intéressantes pour cette étude.

4.3.1 Techniques de vectorisation

Word2vec est une technique mise au point par l'équipe de Tomas Mikolov chez Google [22] qui repose sur un réseau de neurones à deux couches, traitant des n-grams ou des skip-grams. Cette technique a l'avantage de prendre en compte le contexte des mots (n-grams) voir des locutions. Les textes ne sont donc plus considérés comme un sac non ordonné de mots, comme le font les deux autres techniques. Word2vec préserve un peu mieux la sémantique des mots et donc des particularités des textes. Cependant, les indices du vecteur résultant de word2vec ne sont pas reliés à un mot mais à un ensemble de mots, ce qui rend l'interprétation plus ou moins possible. Enfin, word2vec permet le contrôle du nombre de dimensions du vecteur en le fixant.

TF-IDF est une technique de dénombrement et de pondération des mots d'un texte [16]. Généralement elle est appliquée en deux étapes : la première consiste à calculer la fréquence des mots (TF) dans chaque texte, et la deuxième consiste à calculer l'importance de ces mots au sein du corpus (IDF). L'inversion de la technique est parfaitement possible car chaque indice des vecteurs est relié à un unique mot. Cependant, la taille des vecteurs est généralement très grande. Afin de la réduire, j'ai envisagé deux options : calculer TF puis sélectionner un nombre fixe de mots qui ont la plus forte influence sur la variable status ou calculer TF et transformer l'espace des fréquences en un sous espace de taille réduite grâce à une fonction de hash (« hash trick »).

Pour la première option, implémentée sous Spark par CountVectorizer et ChiSelector, l'interprétabilité est conservée car le mapping entre les mots et les indices des vecteurs n'est pas modifié. Par contre, la deuxième option, implémentable sous Spark par HashingTF, interdit toute interprétation car, par définition, les fonctions de hash ne sont pas inversibles.

Seule la première option a été mise en œuvre dans cette étude afin, non seulement de garder possible une interprétation du modèle, mais surtout pour guider la sélection des mots selon une méthode supervisée (χ^2) appliquée au financement des projets, plutôt que de laisser au hasard des collisions de la fonction de hash, la création de regroupement de mots.

Par contre, quelle que soit l'option prise, le nombre de mots sélectionnés pour la première option ou le nombre de dimensions pour l'espace d'arrivée de la fonction de hash pour la seconde option, ces deux nombres doivent être à la fois suffisamment petits pour que le cluster puisse calculer les vecteurs ou les faire tenir en mémoire, et suffisamment grands pour que les composantes des vecteurs ne soient pas majoritairement nulles. Par exemple, en poussant le raisonnement à l'extrême, avec la première option, les textes qui ne sont pas constitués des mots sélectionnés, seront représentés par le vecteur nul.

LDA est un algorithme de modélisation de topics de textes non supervisé [2]. Il propose de résumer un ensemble de textes par un ensemble de topics constitués de mots provenant de l'ensemble des textes. LDA sous entend que les topics, les thèmes abordés par les textes, sont distribués uniformément dans le corpus. Ainsi, les textes sont représentés par des vecteurs dont les composantes sont les poids vis à vis de chacun des topics. Les topics sont eux même des vecteurs dont les composantes sont les poids des mots qui les composent. On fixe le nombre de topics à l'avance ainsi la taille des vecteurs est contrôlée.

De la même façon que pour TF-IDF, le choix du nombre de topics est important : trop petit, les composantes des vecteurs sont proches de zéro, trop grand, la technique n'est pas réalisable sur le cluster. Cette technique n'est pas strictement inversible mais il est possible d'interpréter les topics si l'ensemble des mots qui les composent décrivent un contexte cohérent.

4.3.2 Résultats

L'entraînement d'un modèle Word2vec a été un échec pour cette étude : le cluster n'a pas assez de mémoire. En effet, cette méthode, grande consommatrice en mémoire, est généralement appliquée à de petits corpus. Cependant, il est possible d'utiliser un modèle pré-entraîné notamment celui de Google, or son format n'est toujours pas supporté par Spark : il n'est pas possible de l'importer sous Spark (issue Jira toujours ouverte [33]).

FastText [17] qui est une bibliothèque Python, propose un grand nombre de modèles pré-entraînés [29] sur un corpus de Wikipedia. Cependant, son couplage avec Spark n'a pas été évalué par manque de temps.

Pour TF-IDF, les vocabulaires des textes sont calculés à l'aide de CountVectorizer. Ils sont ensuite sauvegardés afin de garder le mapping mots - indices de vecteur. Puis j'ai fixé le nombre de mots pour la sélection du χ^2 à environ un demi ou un tiers du nombre de mots du vocabulaire des textes. La figure 4.1 en donne les détails.

vocabulaire	nombre de mots	nombre de mots sélectionnés
title	38 030	1 000
statement	44 684	20 000
short_description	44 427	20 000
essay	113 679	30 000

Table 4.1 : Sélection des mots TF-IDF

Cependant l'entraînement d'un modèle ChiSelector sur l'ensemble du dataset (1 million de projets) s'est également soldé par un échec : trop de calculs pour les nœuds du cluster (perte de contrôle). J'ai décidé de sous-échantillonner le dataset afin de diminuer le nombre de calculs (détails section 5.1). Arbitrairement, j'ai fixé à cent mille le nombre de projets. De plus, j'ai décidé d'équilibrer le nombre de projets financés et celui des projets non financés. Cette décision est évaluée dans le chapitre 5.

Grâce au sous-échantillonnage, la vectorisation arrive à son terme. Ces vecteurs sont joints à la table projects afin de former un nouveau dataset (le dataset C, voir 5.1)

Enfin, pour la vectorisation par LDA, configurée avec les hyperparamètres par défaut (automatique), le cluster n'a pu terminer les calculs que pour une centaine de topics. Au-delà, le manque de mémoire, même sur le dataset évoqué au paragraphe précédent, arrête l'entraînement du modèle LDA. Un examen rapide des poids des topics et ceux des vecteurs montre que la plupart d'entre eux sont quasiment nuls. Cette vectorisation a donc été abandonnée.

5

Modélisation

Afin d’aborder la modélisation du financement des projets sur la plate-forme Donors Choose, il est important de faire le point sur les caractéristiques des datasets utilisés. Une fois ce rappel fait, le chapitre continue sur l’Analyse Factorielle de Données Mixtes (AFDM), apportant des directives sur le choix des algorithmes de modélisation. Enfin, il présente la réalisation et les résultats des algorithmes sélectionnés : arbre de décision simple, Random Forest (RF) et Gradient Boosting (GBT).

5.1 Datasets

nom	% non financés	nb variables	vecteurs tf-idf	nb lignes
A	22	46	non	575 174
B	50	46	non	100 020
C	50	50	oui	100 020

Table 5.1 : Compositions des datasets pour la modélisation

Afin de former un ensemble opérationnel de données, il faut réunir par jointure les tables du dataset Kaggle enrichi et les nettoyer. La table 5.1 présente les différents datasets issus de cette préparation. La colonne « % non financés » indique le pourcentage de projets non financés parmi les enregistrements de ces datasets. La colonne « vecteurs tf-idf » indique si les datasets ont les variables issues de la vectorisation des textes des projets (voir 4.3).

La première étape pour la construction de ces datasets, a consisté à nettoyer les tables de leurs incohérences. Par exemple, les projets dont la durée est négative (incohérence des dates de dépôt et d’expiration de projet), les projets en cours (« Live ») et les projets dont le grade_level est « unknown », n’ont pas été retenus. Pour la table teachers, les enregistrements où le préfixe est original, n’ont pas été retenus, et les modalités « Mrs. » et « Ms. » ont été fusionnées ensembles.

La deuxième étape a consisté à joindre entre elles les tables schools, teachers, projects et la table possédant les descripteurs de texte. Le résultat de ces deux étapes conserve à 95 % des enregistrements provenant de la table projects.

La troisième étape a consisté à supprimer toutes les lignes comportant des valeurs manquantes. Suite à cette étape, la perte totale du nombre d’enregistrements s’élève à 48 %. Elle est due principalement aux valeurs manquantes de la variable students_reached. Au vu du nombre de lignes restantes (575 174) et surtout de la distribution de cette variable (A.38), j’ai considéré qu’une imputation n’était pas nécessaire. Quoi qu’il en soit, cette variable s’est avérée importante (voir 5.3).

La quatrième étape a consisté à indexer les modalités des variables qualitatives afin de les préparer aux algorithmes de modélisation et de sauvegarder la correspondance entre les indices et les modalités. Le résultat de cette étape est le dataset A.

Le dataset B est un sous-échantillonnage du dataset A. L'explication du sous-échantillonnage est donnée à la section 4.3.2.

Le rapport entre projets financés et non financés est également ramené à la parité afin d'étudier son influence.

Enfin le dataset C est issu de la vectorisation des textes des enregistrements du dataset B. A noter que `project_count`, `expired_count`, `funded_count`, `county_success_ratio` et `city_success_ratio` ne font pas partie des variables participant à la modélisation. En effet, ces variables sont calculées à partir de la variable `status` qui doit être modélisée. Ces variables masquent l'effet des autres variables et sont pour `city_success_ratio` et les compteurs de projets des professeurs (`_count`), impossibles à calculer, pour les écoles et les professeurs présentant pour la première fois un projet car le succès de ces villes et de ces professeurs ne sont pas encore mesurables. Cependant, `state_success_ratio` est conservée car le pourcentage de succès de financement, au niveau des états, est suffisamment générique pour s'appliquer aux nouveaux participants de la plate-forme.

5.2 AFMD

L'AFDM a un double but. Il s'agit, d'une part, de construire une synthèse commune entre les variables quantitatives et les variables qualitatives des datasets, un codage optimal, afin d'alimenter des algorithmes qui ne prennent en compte que des informations numériques, et d'autre part, de prévoir le succès des algorithmes de classification linéaire.

L'AFDM n'est pas implémentée sous Spark. Selon R. Rakotomalala, l'AFDM est une généralisation de l'ACP et de l'ACM. Il est possible de calculer une AFDM en appliquant une ACP sur les données transformées au préalable [27]. J'ai réalisé une implémentation de l'AFDM sous Spark, en centrant et réduisant les valeurs des variables quantitatives, en appliquant un facteur de normalisation (voir annexe D.1) sur les variables qualitatives encodées selon la méthode one-hot et enfin en exécutant l'implémentation de l'ACP écrite par Spark sur l'ensemble des variables transformées. J'ai appliqué l'AFDM sur le dataset A. Comme les valeurs des variables quantitatives présentent un grand nombre d'outliers, j'ai également expérimenté des transformations sur celles-ci (avant le centrage-réduction). La table 5.2 présente le pourcentage de variance expliquée pour les trois premiers axes factoriels. On observe que les pourcentages sont très faibles et ne sont pas spécialement impactés par les transformations des variables quantitatives.

A la suite de l'étude bidimensionnelle, l'AFDM confirme l'hypothèse de non linéarité de la classification du financement des projets. Sachant que les variables quantitatives ont de nombreux outliers, j'ai considéré que les algorithmes basés sur les arbres de décision, sont de bons candidats pour cette classification. En effet, d'une part ils sont robustes aux outliers et d'autres part ils supportent les variables qualitatives et quantitatives sans transformation (les variables quantitatives sont rendues discrètes par l'hyperparamètre `max bins`). J'ai réalisé l'entraînement d'un modèle classique d'arbre de décision, l'optimisation d'un modèle Random Forest (RF) et celui d'un modèle Gradient Boosting (GBT). Les datasets utilisés sont le dataset A et B. Le dataset C, le seul à posséder les vecteurs des textes, n'a pas été utilisé car le cluster a été incapable d'entraîner les modèles sur le dataset C à cause de dépassements de mémoire dus à un trop grand nombre de dimensions introduites par les vecteurs TF-IDF. De plus, j'ai estimé que l'interprétation de décisions sur des vecteurs TF-IDF avait peu d'intérêt, eu égard à la complexité de sa réalisation.

type de transformation	1er axe	2e axe	3e axe
aucune	0.030	0.023	0.019
log naturel	0.031	0.022	0.019
log 10	0.031	0.022	0.019

Table 5.2 : AFDM : pourcentages de variance expliquée pour les premiers axes factoriels

5.3 Arbre de décision

Bien que cet algorithme soit le moins performant des trois utilisés dans cette étude, il est intéressant d'y avoir recours car il permet de calculer une référence pour la mesure de la performance de classification, de visualiser un arbre illustrant la classification et de présenter les paramètres communs aux trois algorithmes.

hyperparamètres communs	
min instance per node	1
max bins	52
impurity	gini ou entropy
maxDepth	max 30
metric	AUC

Table 5.3 : Configuration de l'arbre de décision

La table 5.3 établit la liste des hyperparamètres de l'entraînement des algorithmes basés sur des arbres de décision. « Min instance per node » est le critère d'arrêt de segmentation des nœuds. Il est par défaut fixé à 1 et cette valeur convient. « Max bins » représente le nombre maximal de modalités pour les variables qualitatives et le nombre de classes pour les variables quantitatives discrétisées. Il est fixé à 52 pour le nombre d'états aux EU. « Impurity » est le critère de segmentation (gini ou entropie). « MaxDepth » représente la profondeur de l'arbre. Il est limité à 30 sous Spark. Comme status ne possède plus que deux modalités (la modalité « Live » a été supprimée), la métrique de performance des prédictions est l'aire sous la courbe de ROC (AUC). Un arbre d'une profondeur de 4 (AUC à 0,61) est donné en exemple en annexe D.1. Ce dernier a été généré à l'aide de la bibliothèque Spark tree plotting [31] et Graphviz [15].

5.4 Random Forest

Les hyperparamètres spécifiques à l'algorithme RF sont décrits à la table 5.4. « FeatureSubsetStrategy » représente la fonction d'échantillonnage des variables. La fonction racine carrée est généralement conseillée pour la classification selon Breiman [3]. De plus, elle permet de moins consommer de mémoire en sélectionnant moins de variables par arbres. « NumTrees » précise le nombre d'arbres à entraîner par l'algorithme. Enfin, « Dataset split » qui n'est pas un hyperparamètres de RF, est le ratio de partition du dataset pour l'entraînement du modèle (80 % des enregistrements) et pour la caractérisation de ses performances (20 %).

Afin d'optimiser l'algorithme, j'ai étudié l'influence de ces hyperparamètres sur la métrique de performance de prédiction. Le protocole est simple, il s'agit de mesurer l'influence des hyperparamètres individuellement, les autres étant fixés, puis de trouver les optimums par dichotomie. Plutôt que d'utiliser les méthodes grid search, random search ou l'optimisation bayésienne, j'ai préféré

Hyperparamètres Random Forest	
featureSubsetStrategy	sqrt
numTrees	max 250
dataset split	80/20 %

Table 5.4 : Hyperparamètres spécifiques à Random Forest

utiliser ce protocole car le cluster étant très limité en terme de mémoire, il était nécessaire de découvrir les bornes des intervalles de valeurs des hyperparamètres compatibles avec le cluster. Or je n'ai pas trouvé pratique les méthodes citées pour calculer ces intervalles : lors d'un dépassement de mémoire, leur implémentation ne donne pas la valeur fautive de l'hyperparamètre. Concernant la validation, j'ai choisi une approche classique avec trois jeux de données (entraînement, validation et test) car la validation croisée est trop gourmande en mémoire et en temps de calcul. La table 5.5 résume la configuration de l'optimisation commune à RF et GBT.

Optimisation	
TrainValidationSplit	
parallel factor	1
dataset entraînement	64 %
dataset validation	16 %
dataset test	20 %

Table 5.5 : Configuration de l'optimisation

Afin de trouver un jeu optimum de valeurs d'hyperparamètre, j'ai conduit une série d'entraînement sur le dataset B car il est moins volumineux que le dataset A. La figure 5.1 montre l'influence de la profondeur de l'arbre pour les deux critères de segmentation (le score représenté est celui de la validation). Naturellement, plus l'arbre est profond, plus le score AUC augmente, avec un pallier dans les environs de 18. Le score de test (non représenté) confirme les bonnes performances et une généralisation du modèle (overfitting évité). Concernant le critère de segmentation, il semble que gini et entropie se valent. La figure 5.2 illustre l'influence du nombre d'arbres, toujours pour les deux critères. Son profil logarithmique, montre un gain d'AUC faible à partir de 50 arbres. On note également que le critère entropie donne systématiquement de moins bons scores.

Les entraînements au-delà des valeurs indiquées sur les figures n'aboutissent pas à cause du manque de mémoire. D'après les résultats de ces expériences, j'ai trouvé un optimum par dichotomie réalisable sur le cluster (profondeur 30 ; nombre d'arbres à 50 ; critère gini) pour le dataset B, avec un score de 0,9852 (figure récapitulative, voir 5.6). L'optimum pour le dataset A est obtenu avec moins de profondeur d'arbre (25), et moins d'arbres (10), toujours en raison de la limitation en mémoire du cluster, avec un score AUC de 0,962. En entraînant un modèle sur le dataset B mais avec les mêmes valeurs d'hyperparamètres que précédemment, le score s'élève à 0,963. Il semble que l'équilibrage entre les proportions de projets financés et non financés n'ait pas d'influence sur l'algorithme RF. Enfin, la figure 5.3 donne l'importance des douze premières variables exprimées en pourcentage. Les trois plus importantes variables pour la classification sont le coût du projet, le ratio coût du projet par élèves et le nombre d'élèves concernés par le projet. A noter que les autres variables sont presque uniquement des descripteurs de texte, sauf une qui provient du jeu de variables extérieures (zip_med_home_value).

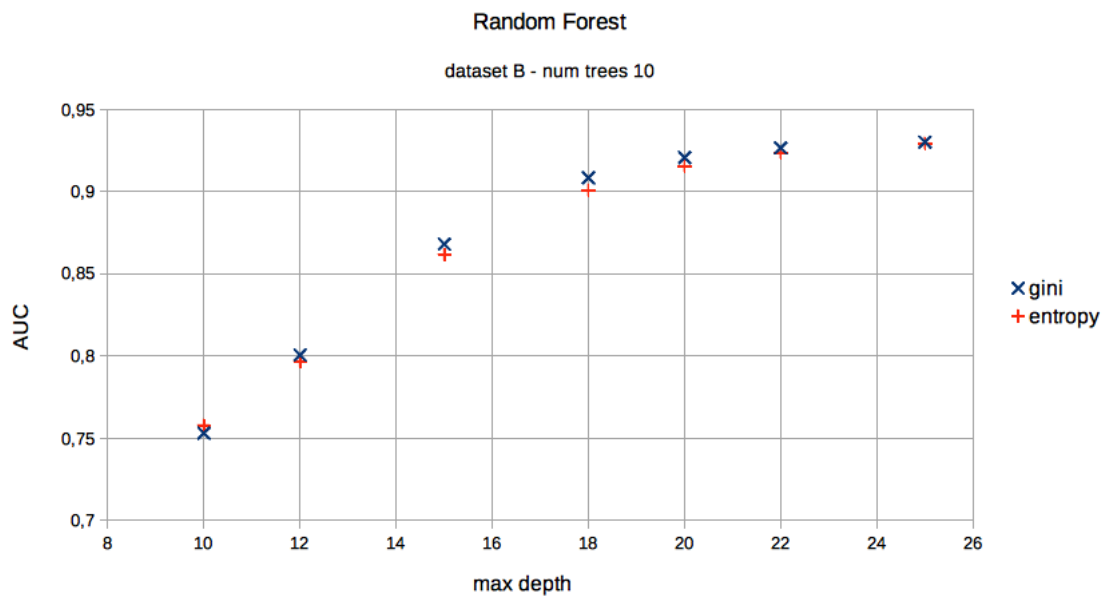


Figure 5.1 : Optimisation de la profondeur d'arbre (RF)

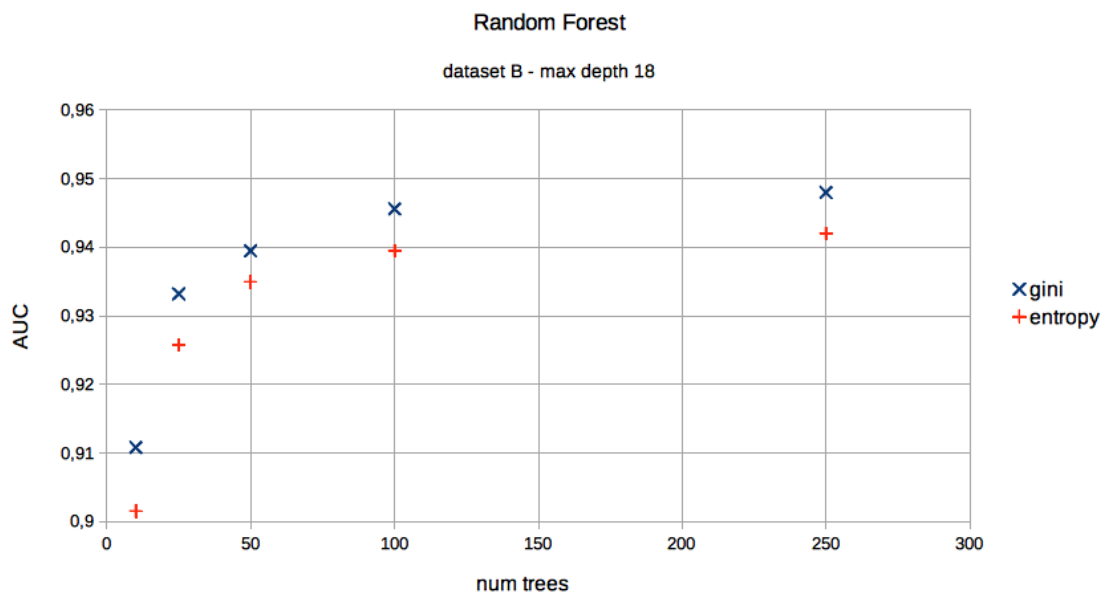


Figure 5.2 : Optimisation du nombre d'arbres (RF)

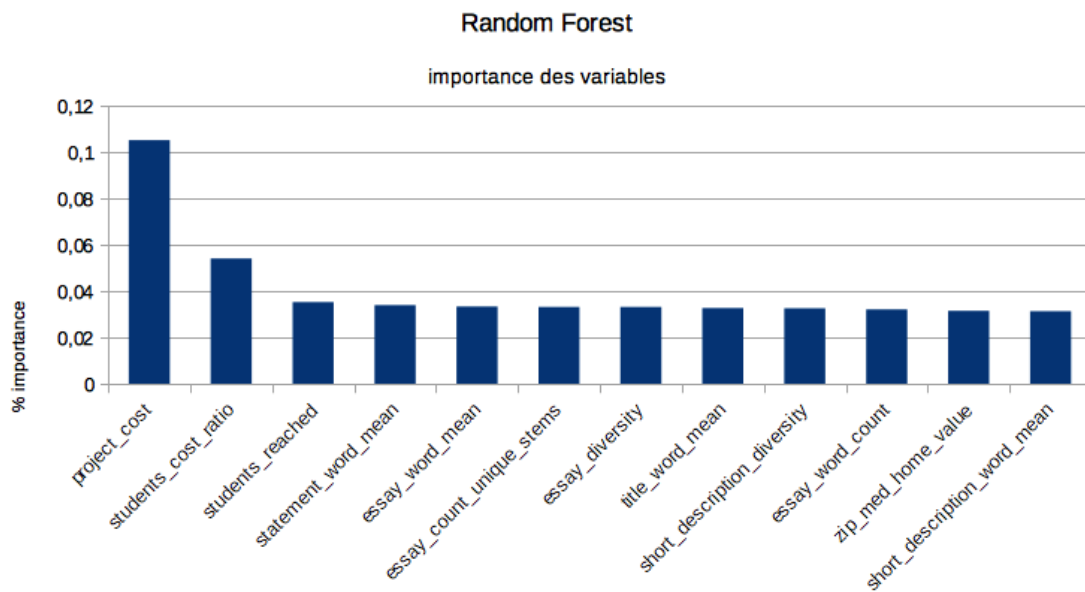


Figure 5.3 : Importance des 12 premières variables (RF)

5.5 Gradient Boosting

Les hyperparamètres spécifiques à l'algorithme GBT sont décrits à la table 5.6. « FeatureSubsetStrategy » a la même signification que l'hyperparamètre RF du même nom. « MaxIter » est le nombre maximal d'itérations sur le jeu de données en entrée de modèle. « stepSize » est l'incrément d'apprentissage et sa valeur par défaut convient (0,1). Enfin, « Dataset split » a également la même fonction que dans la mesure de performance du modèle RF.

Hyperparamètres Gradient Boosting	
featureSubsetStrategy	sqrt
maxIter	max 75
stepSize	0.1
dataset split	80/20 %

Table 5.6 : Hyperparamètres Gradient Boosting

A l'image de l'optimisation de RF, j'ai étudié l'influence des hyperparamètres de GBT. Il s'agit de mesurer uniquement celle de « maxIter », la valeur par défaut de « stepSize » convenant bien. La figure 5.4 montre que plus GBT itère sur le dataset, plus le score AUC augmente. Comme pour RF, l'optimum de GBT réalisable sur le cluster est trouvé par essais-échecs et dichotomie. Pour le dataset B, il est obtenu avec une profondeur de 30 et 20 itérations et un score AUC de 0,9857. Concernant le dataset A, le score AUC est de 0,949 avec une profondeur de 25 et 5 itérations. L'importance des variables obtenues par GBT, illustrées à la figure 5.5 ne change pas radicalement de celle par RF. Les deux plus importantes variables restent le coût du projet et le ratio coût du projet par élève. L'ordre des autres variables change mais il n'a pas beaucoup d'importance car les valeurs de l'importance des variables sont très proches les unes des autres.

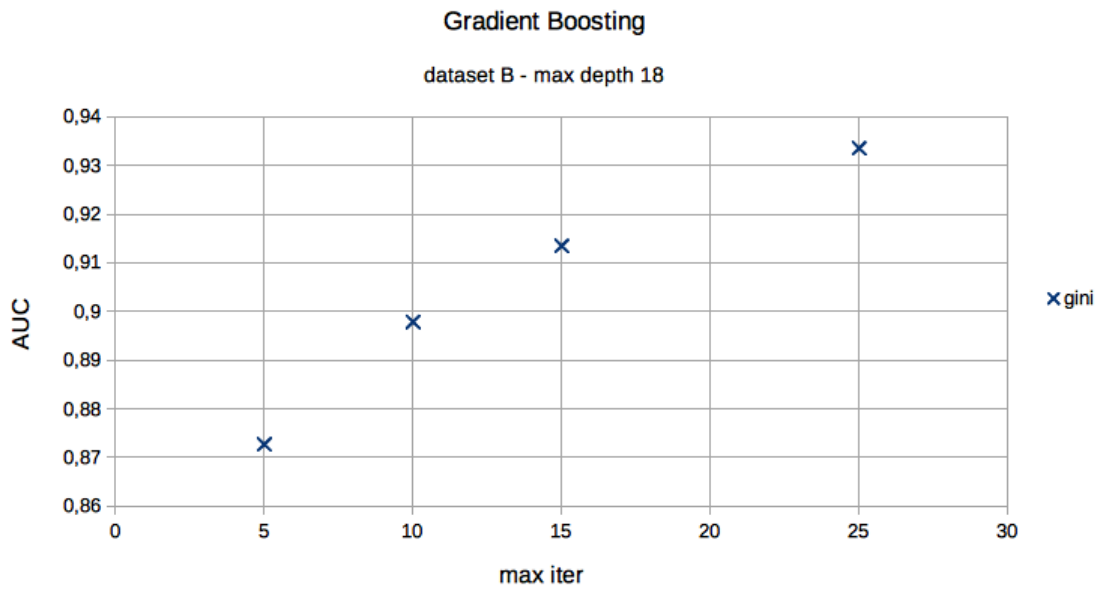


Figure 5.4 : Optimisation du nombre d'itérations (GBT)

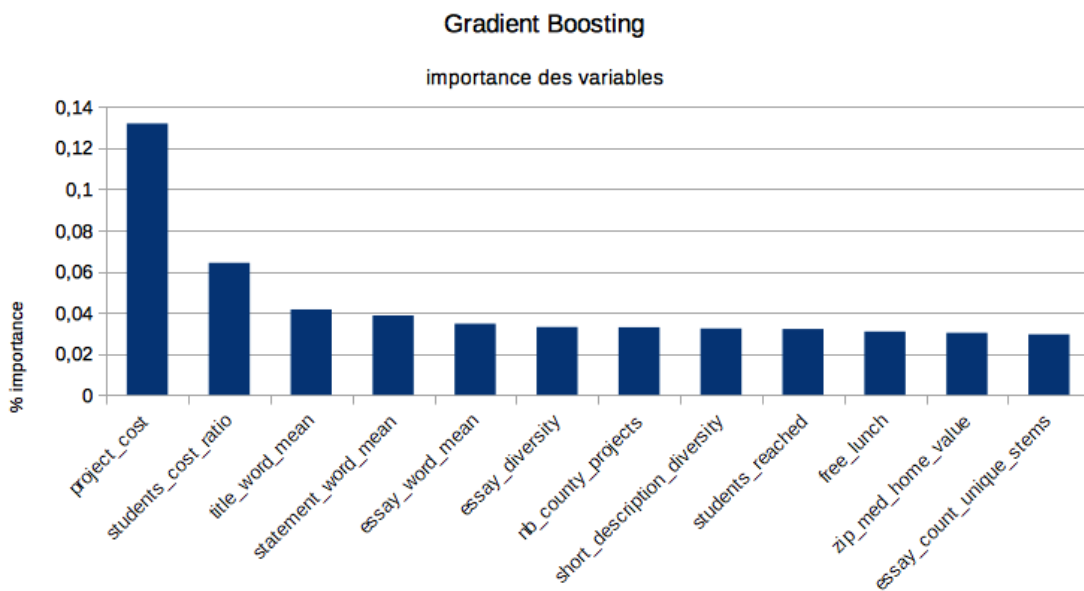


Figure 5.5 : Importance des 12 premières variables (GBT)

5.6 Récapitulatif

La figure 5.6 donne un récapitulatif des scores obtenus par les différentes techniques de modélisation. On constate que RF est préférable non seulement parce que l'on obtient un meilleur score sur un plus grand volume de données, mais surtout parce que son entraînement est bien plus rapide que GBT (environ deux heures pour le dataset A). En effet, GBT entraîne les arbres en série alors que RF les entraîne en parallèle.

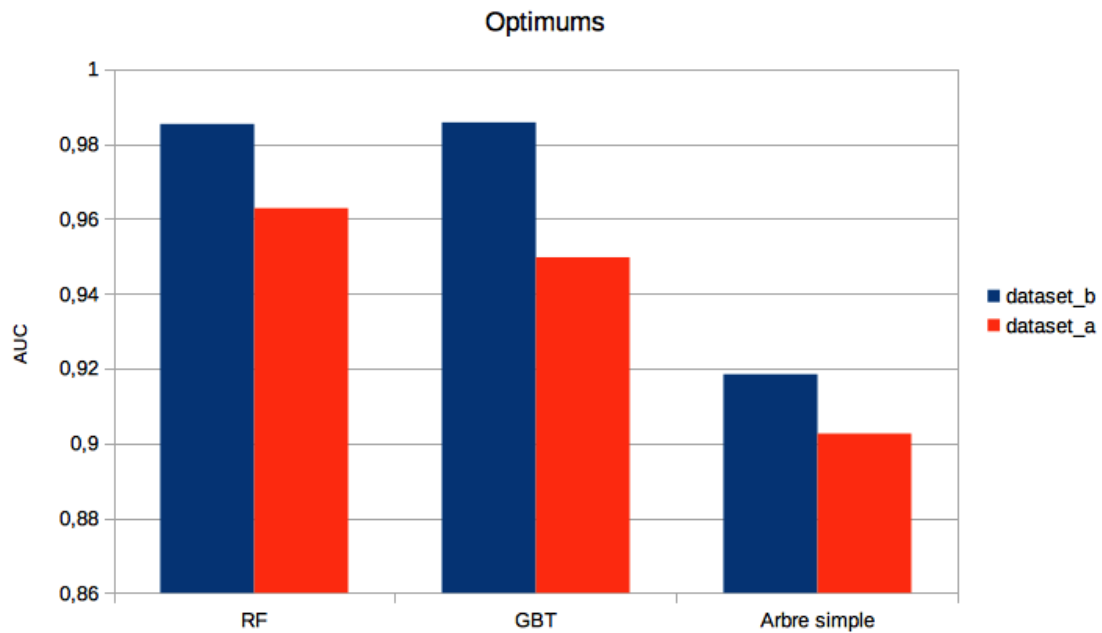


Figure 5.6 : Score AUC pour les optimums trouvés

6

Conclusion

Donors Choose est une plate-forme de financement participatif de projets proposés par des professeurs des écoles aux États Unis d'Amérique. Cette plate-forme est intéressante car elle offre à l'analyse des données relativement riches et volumineuses. Cette étude propose de comprendre ces données, d'expliquer et de modéliser le financement de ces projets. Afin de réaliser les calculs nécessaires, elle s'appuie sur un cluster Spark de trois nœuds, créé à cette fin et couplé à la base MongoDB. La scalabilité de ce cluster est abordée.

Le jeu de données utilisé pour cette étude est le résultat de différentes jointures entre deux des trois datasets proposés par Donors Choose et Kaggle. Il est également enrichi par des variables extérieures provenant du recensement américain et des variables calculées réglant des incohérences dans la structure des jeux de données, ou rendant compte de cardinalités multiples entre les tables de données.

Les études, uni et bidimensionnelle, apportent un éclairage sur Donors Choose, notamment sur la relation de proximité géographique entre les donateurs et les écoles à l'initiative des projets. Elles révèlent non seulement que les variables quantitatives présentent de nombreux outliers, mais aussi que leur distribution a un profil très étalé, et que certaines variables qualitatives ont de très nombreuses modalités, parmi lesquelles certaines s'avèrent inintéressantes. Elles montrent également une très faible corrélation entre la variable de financement et les autres variables. La nature non linéaire du problème de modélisation du financement des projets est confirmée par une implémentation maison de l'analyse factorielle de données mixtes.

Les projets Donors Choose comportent des textes, c'est pourquoi une partie de ce travail a porté leur vectorisation et la création de descripteurs. La racinisation et la lemmatisation nécessaires, pour l'une au calcul des descripteurs de texte, et pour l'autre à la vectorisation, ont été réalisées à l'aide de Spark NLP. Il s'agit d'une bibliothèque assez jeune mais totalement intégrée à Spark et à son modèle de calcul distribué. La vectorisation des textes est obtenue par transformation TF-IDF et par sélection des mots par test du χ^2 , afin de contraindre le nombre de dimensions des vecteurs selon la variable de financement. Une alternative de vectorisation par Latent Dirichlet Allocation a été explorée, mais sans succès : les composantes des vecteurs sont quasiment nulles à cause du trop faible nombre de topics induit par les limitations du cluster.

Le choix des algorithmes pour modéliser le financement des projets Donors Choose s'est porté sur ceux basés sur les arbres de décision. En effet, ces algorithmes sont adaptés aux caractéristiques de cette étude : de nombreux outliers pour les variables quantitatives et une nature non linéaire pour le financement des projets. Ce travail a cherché à optimiser des modèles Random Forest et Gradient

Boosting par validation classique (entraînement - validation - test). Sur un jeu de données exempt de vecteurs TF-IDF, le meilleur modèle, obtenu par Gradient Boosting, donne un score AUC de 0,9857 pour la prédiction du financement des projets. Par manque de temps, le jeu de données issu de la projection sur tous les axes factoriels de l'AFDM et celui comportant les vecteurs TF-IDF n'ont pas pu être exploités.

A la suite de ce travail, il serait intéressant de mieux prendre en compte la particularités des textes des projets, notamment en exploitant les vecteurs TF-IDF, par exemple à l'aide d'un perceptron. En alternative de TF-IDF, l'utilisation d'un modèle pré-entraîné word2vec serait également intéressante. Côté infrastructure de cluster, la mise en place d'une architecture scalable constituerait une bonne expérience, avec, d'une part, Mesos et Yarn pour le partage des ressources matérielles et l'orchestration de tâches, et d'autre part Marathon pour le déploiement et le redémarrage des services (Spark et MongoDB). Enfin, la mise en production du prédicteur ferait une excellente extension à cette étude, avec une possibilité de prédiction et de mise à jour du modèle en continu (streaming).

ANNEXES

A

Variables

A.1 Exemple de page d'un projet

\$1,193 GOAL THIS PROJECT EXPIRED ON AUGUST 31, 2018.

We Need 3D Printers for Our STEAM Lab!

My students need 3D printers in order to create real-world technology experiences for our STEAM lab!


My Students

The student body at my school is extremely diverse. The majority of our families are living in poverty, and nearly half of our students are originally from other countries. Many of my students have few if any books at home, so visiting the school library becomes an integral part of their lives. My students are curious, hardworking and love to be challenged. When we get out equipment such as iPads, I see their faces light up with excitement and anticipation.

My Project

Over the last couple of years, we have been expanding our library into a full STEAM lab. STEAM (science,technology,engineering,art,math) is all about real-world, project-based learning. Our students use tablets, computers, robots and more every day in our library. By expanding our offerings, we are creating 21st century learners. My goal is to introduce my students to the experiences that await them in middle and high school, and eventually college and careers.

With these 6 3D printers, our students will be able to work in small groups to explore virtual design and production software and see their designs become tangible objects.



Mr. Highley
Grades 3-5

Fairdale Elementary School
Fairdale, KY
More than three-quarters of students from low-income households

This project will reach 600 students.

Fairdale, KY Grades 3-5

More than three-quarters of students from low-income households

College & Career Prep Extracurricular

SHARE MR. HIGHLEY'S PROJECT

Figure A.1 : Description d'un projet

Where Your Donation Goes


MATERIALS	COST	QUANTITY	TOTAL
HopeWant Desktop 3D Printer STEAM for design Mini 3D Printer Kit with 250g PLA Filament TF Card High Accuracy 3D Print Education Windows/MAC/LINUX Supported • AMAZON BUSINESS	\$161.49	6	\$968.94
Materials cost			\$968.94
Vendor shipping charges			FREE
State sales tax			\$0.00
3rd party payment processing fee ?			\$14.53
Fulfillment labor & materials ?			\$30.00
Total project cost ?			\$1,013.47
Suggested donation to help DonorsChoose.org reach more classrooms ?			\$178.85
Total project goal ?			\$1,192.32

Our team works hard to negotiate the best pricing and selections available. ?

Figure A.2 : Détails sur le coût du projet

Project Activity


If you donated to this project, you can [sign in](#) to leave a comment for Mr. Highley.

AUG 22  **Susan Henderson** from Colorado gave

What a great project. Good luck!

The Bill & Melinda Gates Foundation matched this donation

The Bill & Melinda Gates Foundation is pleased to join you in supporting teachers and school communities. Together, we can make this Back to School season the best one yet.

JULY 27  **A donor** from Kentucky gave with help from **Google**


JUNE 4  **A donor** from Kentucky gave with help from **Google**

Figure A.3 : Activités autour du projet

A.2 Description des variables

En orange : les variables empruntées au dataset donors choose.

En bleu : les variables calculées.

nom de variable	sémantique	type
_id	identifiant de l'école	hash
school_name	nom de l'école	texte
school_type	type d'école	nominal
free_lunch	ratio de repas subventionnés	quanti
school_state	état de l'école	nominal
school_zip	code postal de l'école	code
school_county	comté de l'école	nominal
school_district	district de l'école	nominal
school_city	ville de l'école	nominal
poverty_level	niveau de pauvreté	ordinal
school_ncesid	code NCESID de l'école	code
grade_level	niveau de classe	ordinal
school_charter	est ce une école autonome	binaire
school_charter_ready_promise	?	binaire
school_kipp	est ce une école KIPP	binaire
school_magnet	est ce une école subventionnée	binaire
school_nlns	est ce une école « New Leaders »	binaire
school_year_round	le calendrier scolaire est-il réoga	binaire
school_latitude	latitude localisation	coordonnées
school_longitude	longitude localisation	coordonnées
zip_density	density en habitants/miles ²	quanti
zip_med_home_value	médiane valeur habitation en \$	quanti
zip_med_income	médiane revenus en \$	quanti

Table A.1 : Variables de la table schools

nom de variable	sémantique	type
_id	identifiant de projet	hash
school_id	foreign key schools	hash
teacher_id	foreign key teachers	hash
status	état de financement	nominal
project_cost	coût du projet	quanti
project_type	porteur du projet	nominal
teacher_project_seq	ordre du projet lors don	ordinal
grade_level	niveau classe	ordinal
title	titre du projet	texte
statement	engagement du projet	texte
short_description	courte description	texte
essay	longue description ou essais	texte
subject_cat	catégorie du projet	nominal
subject_subcat	sous catégorie du projet	nominal
resource_cat	catégorie de ressources	nominal
posted_date	date de soumission	timestamp
expiration_date	date d'expiration	timestamp
funded_date	date de financement	timestamp
primary_focus_subject	catégorie du projet	nominal
primary_focus_area	domaine du projet	nominal
secondary_focus_subject	sous catégorie du projet	nominal
secondary_focus_area	sous domaine du projet	nominal
students_reached	nombre d'élèves impactés	quanti
students_cost_ratio	ratio students_reached/project_cost	quanti
posted_month	n° du mois de soumission	ordinal
expiration_month	n° du mois d'expiration	ordinal
funded_month	n° du mois de financement	ordinal
title_chars	nombre de symboles dans le titre	quanti
statement_chars	nombre de symboles dans l'engagement	quanti
short_description_chars	nombre de symboles dans la courte desc	quanti
essay_chars	nombre de symboles dans l'essais	quanti
duration	durée du projet (expiration_date – posted_date)	quanti
first_project	est ce le premier projet de l'enseignant	binaire
nb_state_projects	nombre de projets dans l'état de l'école	quanti
nb_county_projects	nombre de projets dans le conté de l'école	quanti
nb_city_projects	nombre de projets dans la ville de l'école	quanti
state_success_ratio	ratio de succès dans l'état de l'école	quanti
county_success_ratio	ratio de succès dans le conté de l'école	quanti
city_success_ratio	ration de succès dans la ville de l'école	quanti
nb_state_donors	nombre de donateurs dans l'état de l'école	quanti
nb_city_donors	nombre de donateurs dans la ville de l'école	quanti

Table A.2 : Variables de la table projects

nom de variable	sémantique	type
_id	identifiant du don	hash
project_id	foreign key du projet	hash
donation_id	identifiant du don	hash
donor_id	foreign key du donateur	hash
has_donation_optional	le don comporte t-il un versement à donors choose	binaire
donation_amount	montant du don	quanti
donation_seq	l'ordre du projet dans le panier	ordinal
donation_date	date du don	timestamp
month	n° du mois du don	ordinal

Table A.3 : Variables de la table donations

nom de variable	sémantique	type
_id	identifiant de la ressource	hash
project_id	foreign key projects	hash
resource_name	nom de la ressource	text
quantity	quantité de la ressource	quanti
unit_price	prix à l'unité	quanti
vendor_name	fournisseur	nominal

Table A.4 : Variables de la table resources

nom de variable	sémantique	type
_id	identifiant du prof	hash
teacher_prefix	titre du prof	nominal
fixed_teacher_prefix	titre du prof corrigé	nominal
teacher_first_project_date	date du premier projet du prof	timestamp
teacher_teach_for_america	est-il de « teach for america »	binaire
teacher_ny_teaching_fellow	est-il de « ny teaching fellow »	binaire
project_count	nombre de projets soumis	quanti
live_count	nombre de projets actifs	quanti
expired_count	nombre de projets expirés	quanti
funded_count	nombre de projets financés	quanti
teacher_success_ratio	ratio funded_count/(project_count – live_count)	quanti

Table A.5 : Variables de la table teachers

nom de variable	sémantique	type
_id	identifiant du donateur	hash
donor_city	ville du donateur	nominal
donor_state	état du donateur	nominal
is_teacher	est-il un professeur	binaire
donor_zip	code postale du donateur	code

Table A.6 : Variables de la table donors

A.3 Résumés des variables quantitatives

nom de variable	project_cost	students_reached	students_cost_ratio	title_chars
moyenne	741.52	93.13	23.96	31.91
écart type	1083.26	154.45	57.07	13.06
min	35.29	1.00	0.11	2.00
q1	335.16	22.00	6.24	22.00
q2	515.45	30.00	14.07	30.00
q3	867.53	95.00	27.40	40.00
max	255737.67	1400.00	9104.22	149.00
kurtosis	6070.41	12.86	3993.92	-0.12
skewness	46.14	3.41	44.40	0.61

Table A.7 : Variables quantitatives de la table projects 1/3

nom de variable	statement_chars	short_description_chars	essay_chars	duration
moyenne	121.39	195.40	1720.84	116.64
écart type	47.34	6.89	469.90	15.18
min	16.00	1.00	30.00	-486.00
q1	84.00	194.00	1345.00	118.00
q2	116.00	196.00	1626.00	120.00
q3	157.00	198.00	2015.00	121.00
max	1610.00	399.00	32756.00	489.00
kurtosis	4.59	362.32	19.76	32.43
skewness	0.72	-16.46	1.14	-4.17

Table A.8 : Variables quantitatives de la table projects 2/3

nom de variable	state_success_ratio	county_success_ratio	city_success_ratio
moyenne	0.77	0.77	0.77
écart type	0.04	0.06	0.09
min	0.69	0.00	0.00
q1	0.75	0.74	0.73
q2	0.77	0.77	0.78
q3	0.80	0.82	0.83
max	0.87	1.00	1.00
kurtosis	-0.43	2.99	6.54
skewness	-0.11	-0.62	-1.37

Table A.9 : Variables quantitatives de la table projects 3/3

nom de variable	donation_amount
moyenne	60.66
écart type	166.89
min	0.01
q1	14.78
q2	25.00
q3	50.00
max	60000.00
kurtosis	5664.62
skewness	40.95

Table A.10 : Variables quantitatives de la table donations

nom de variable	unit_price	quantity
moyenne	53.41	2.81
écart type	186.32	8.86
min	0.00	0.00
q1	7.26	1.00
q2	14.39	1.00
q3	36.4	2.00
max	97085.5	4125.00
kurtosis	26893.28	12338.48
skewness	79.75	57.00

Table A.11 : Variables quantitatives de la table resources

nom de variable	free_lunch	zip_density	zip_med_home_value	zip_med_income
moyenne	58.55	3861.60	203523.12	53696.72
écart type	25.50	9886.02	143751.84	21271.31
min	0.00	0.064	9999.00	9475.00
q1	40.00	158.00	108300.00	38979.00
q2	61.00	1023.00	158800.00	49000.00
q3	80.00	3701.00	249600.00	64054.00
max	100.00	141546.00	997000.00	207262.00
kurtosis	-0.87	54.03	5.07	2.50
skewness	-0.34	6.54	2.02	1.26

Table A.12 : Variables quantitatives de la table schools

nom de variable	project_count	live_count	expired_count
moyenne	33.94	0.73	5.67
écart type	557.22	11.65	68.02
min	1.00	0.00	0.00
q1	1.00	0.00	0.00
q2	2.00	0.00	0.00
q3	5.00	0.00	1.00
max	118400.0	1491.00	8520.00
kurtosis	10724.35	4157.69	3281.62
skewness	79.76	51.25	45.86

Table A.13 : Variables quantitatives de la table teachers 1/2

nom de variable	funded_count	teacher_success_ratio
moyenne	26.92	0.73
écart type	512.16	0.38
min	0.00	0.00
q1	1.00	0.5
q2	1.00	1.00
q3	4.00	1.00
max	117660.00	1.00
kurtosis	13892.85	-0.40
skewness	92.18	-1.10

Table A.14 : Variables quantitatives de la table teachers 2/2

A.4 Résumés des variables qualitatives

nom de variable	nombre modalités	modalité majoritaire	pourcentage
status	3	Fully Funded	74.48
project_type	2	Teacher-Led	98.39
grade_level	5	Grades PreK-2	38.91
subject_cat	52	Literacy & Language	22.56
subject_subcat	433	Literacy, Mathematics	8.40
resource_cat	18	Supplies	34.78
primary_focus_subject	29	null	40.67
primary_focus_area	8	null	40.67
secondary_focus_subject	29	null	59.89
secondary_focus_area	8	null	59.89
posted_month	12	8	13.89
expiration_month	12	0	13.96
funded_month	12	null	25.51
first_project	2	false	69.10

Table A.15 : Variables qualitatives de la table projects

nom de variable	nombre modalités	modalité majoritaire	pourcentage
has_donation_optional	2	Yes	85.36
month	12	7	12.30

Table A.16 : Variables qualitatives de la table donations

nom de variable	nombre modalités	modalité majoritaire	pourcentage
vendor_name	32	Amazon Business	44.42

Table A.17 : Variables qualitatives de la table resources

nom de variable	nombre modalités	modalité majoritaire	pourcentage
school_type	5	suburban	31.49
school_state	51	California	11.58
school_county	1 783	Los Angeles	2.89
school_district	10 852	New York City Dep	2.35
school_city	10 400	New York City	2.80
poverty_level	5	highest poverty	40.48
grade_level	5	Grades PreK-2	29.09
school_charter	2	false	83.51
school_charter_ready_promise	2	false	90.50
school_kipp	2	false	93.38
school_magnet	2	false	85.84
school_nlns	2	false	90.26
school_year_round	2	false	88.86

Table A.18 : Variables qualitatives de la table schools

nom de variable	nombre modalités	modalité majoritaire	pourcentage
teacher_prefix	7	Mrs	50.1
fixed_teacher_prefix	6	Mrs	86.34
teacher_teach_for_america	2	false	69.24
teacher_ny_teaching_fellow	2	false	70.73

Table A.19 : Variables qualitatives de la table teachers

nom de variable	nombre modalités	modalité majoritaire	pourcentage
donor_city	15 205	null	10.03
donor_state	52	California	13.88
is_teacher	2	No	89.99

Table A.20 : Variables qualitatives de la table donors

A.5 Répartition des modalités des variables qualitatives

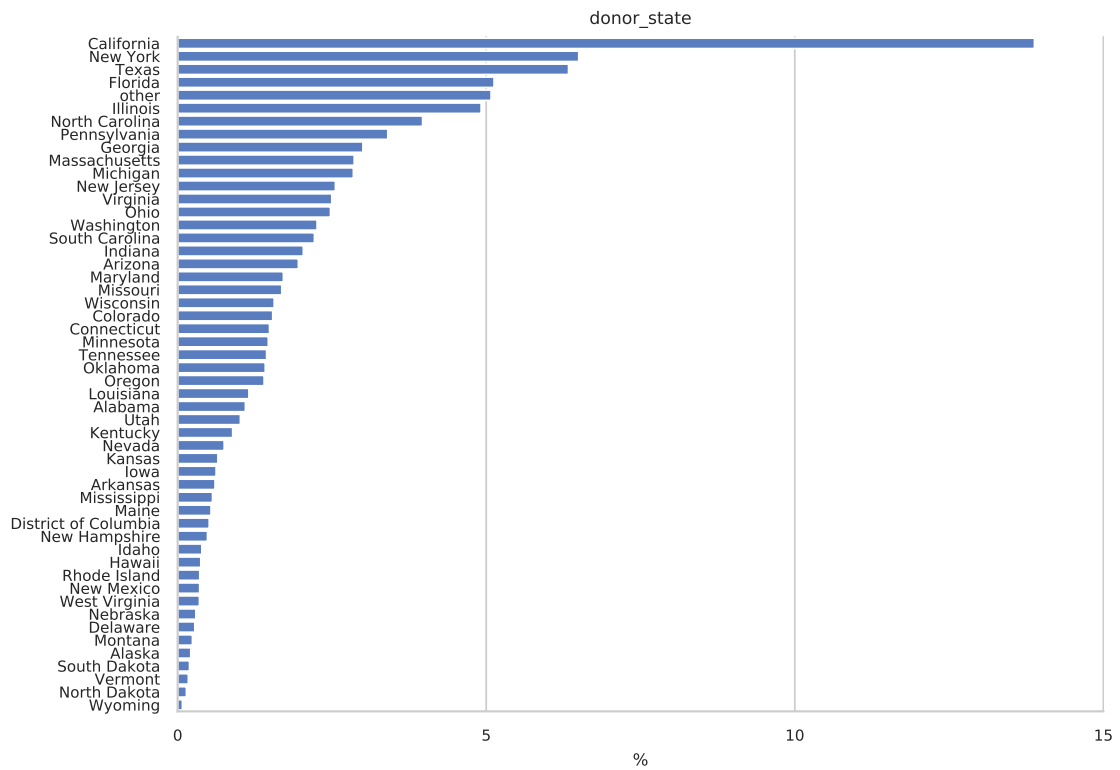


Figure A.4 : Répartition des donateurs par états

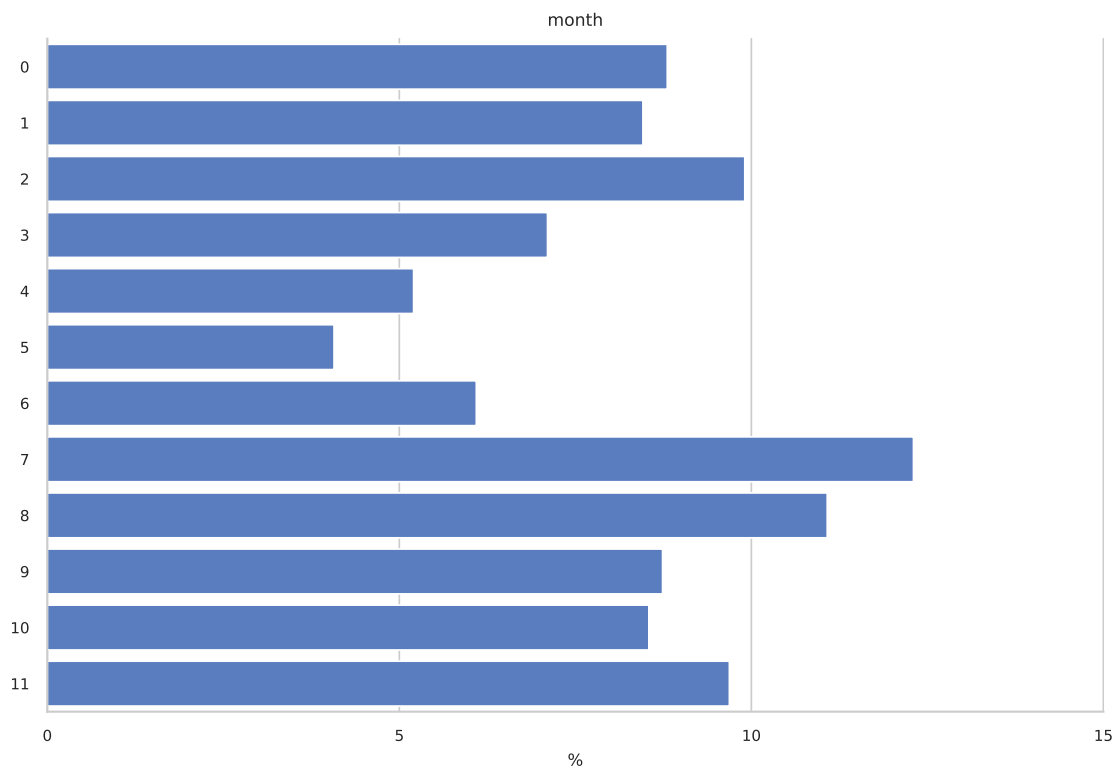


Figure A.5 : Répartition des dons par n° de mois

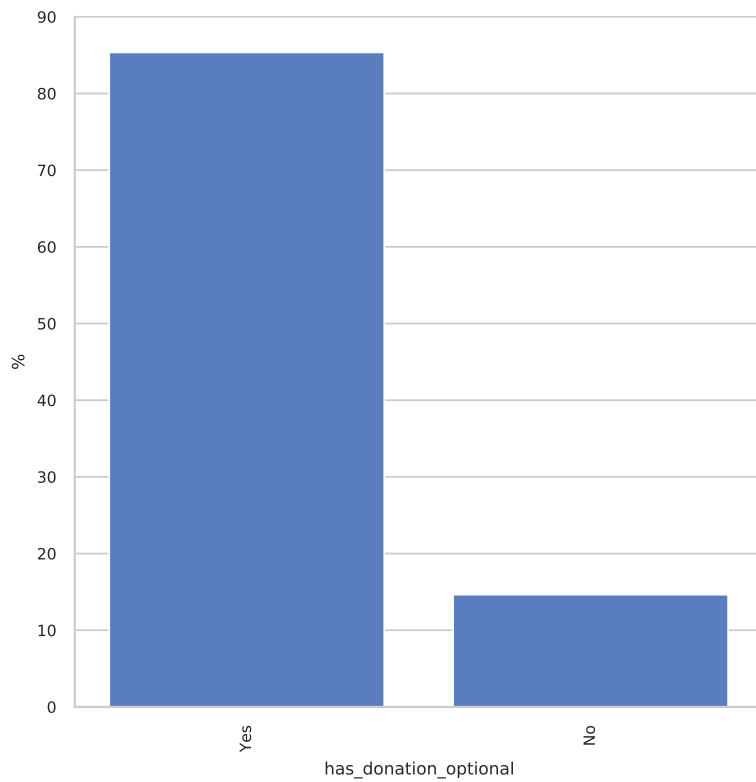


Figure A.6 : Répartition du don optionnel

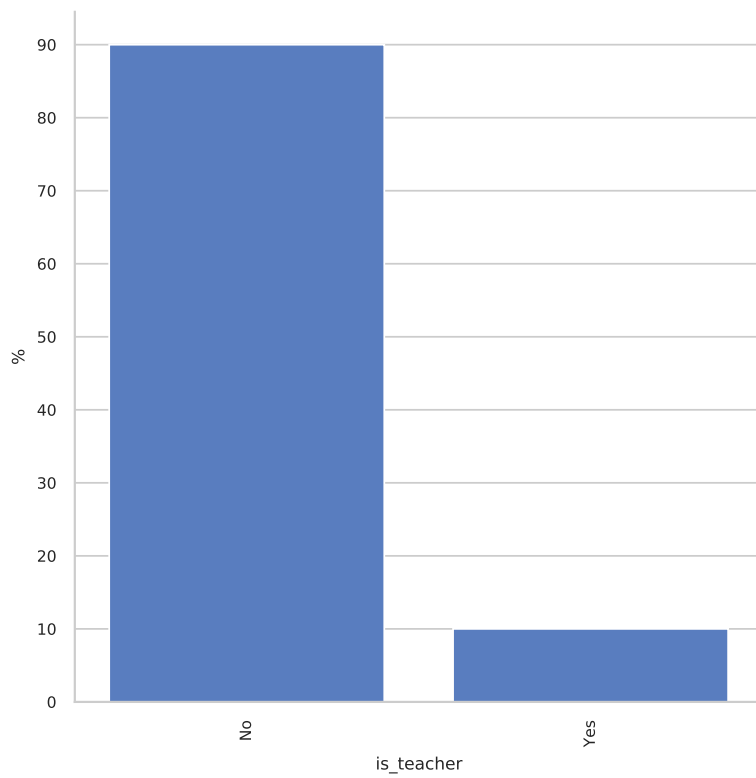


Figure A.7 : Répartition des professeurs parmi les donateurs

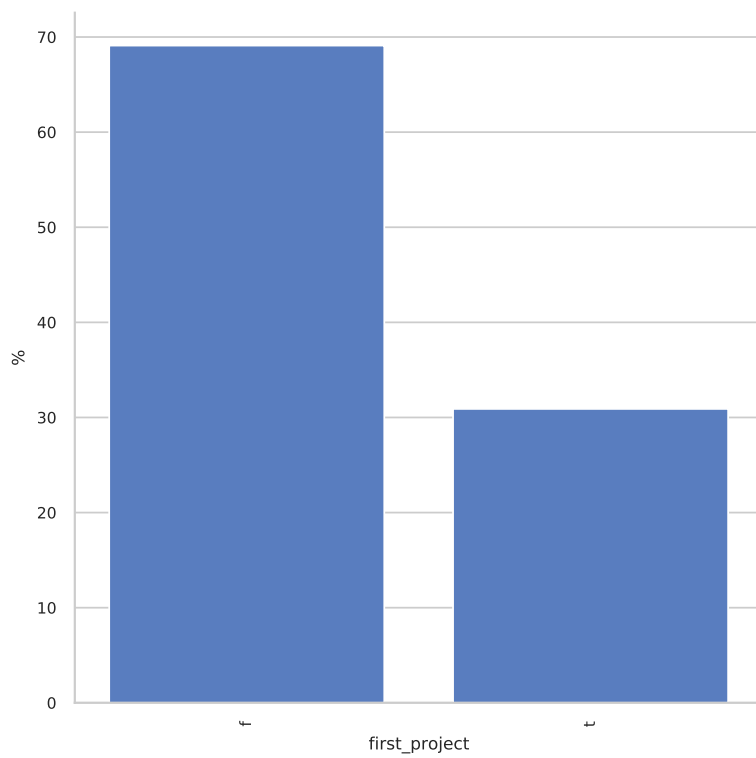


Figure A.8 : Répartition des premiers projets

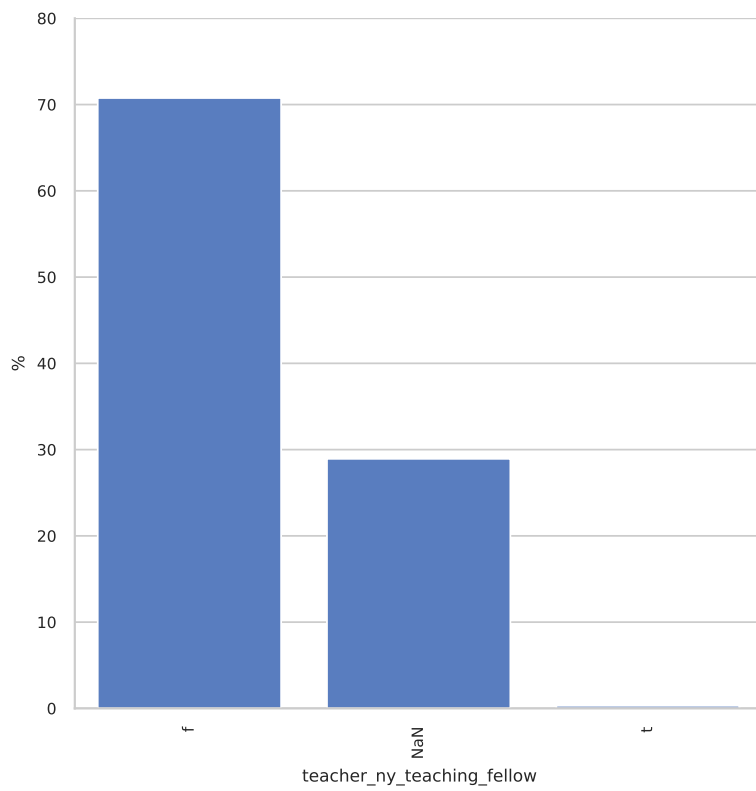


Figure A.9 : Répartition des professeurs "ny teaching fellow"

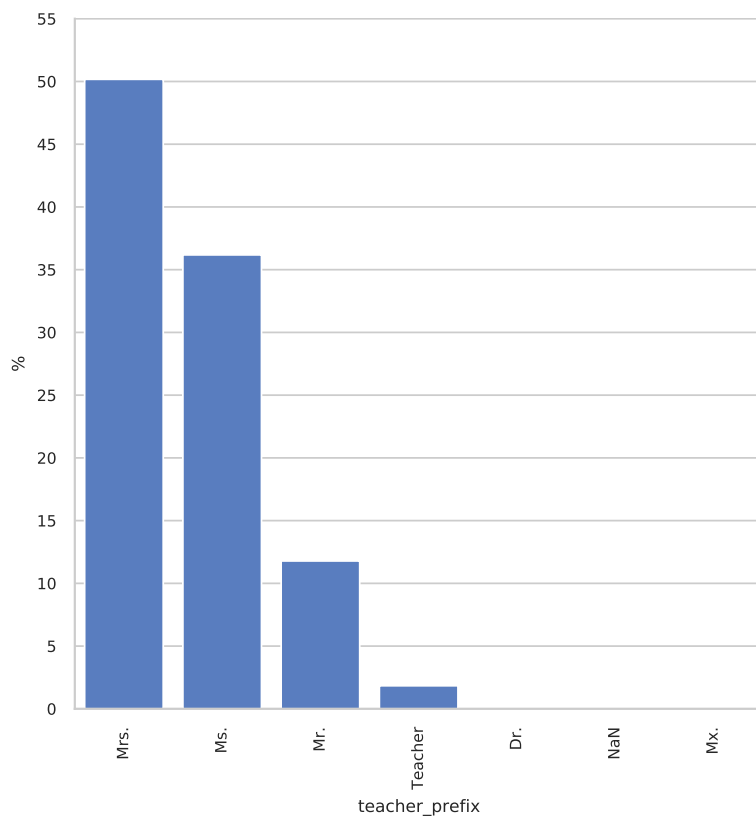


Figure A.10 : Répartition des professeurs par leur préfixes

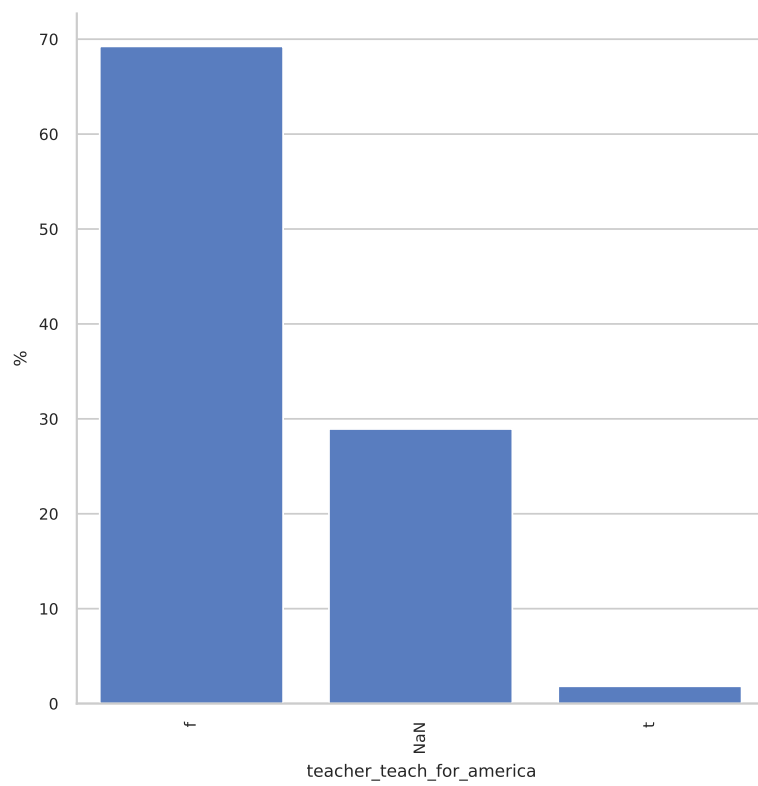


Figure A.11 : Répartition des professeurs "teach for america"

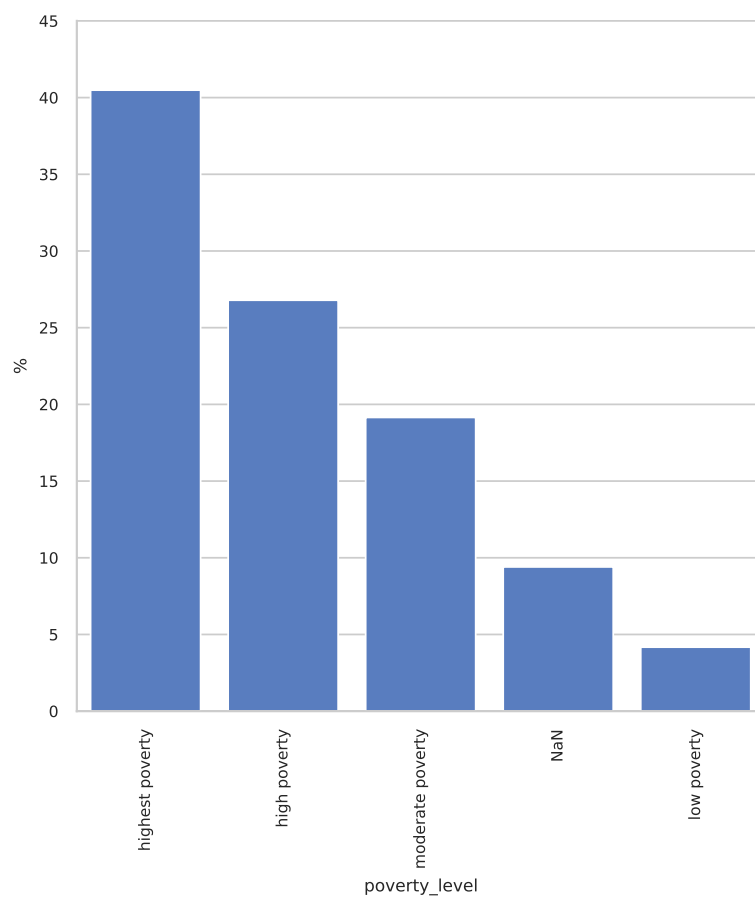


Figure A.12 : Répartition des écoles par niveaux de pauvreté

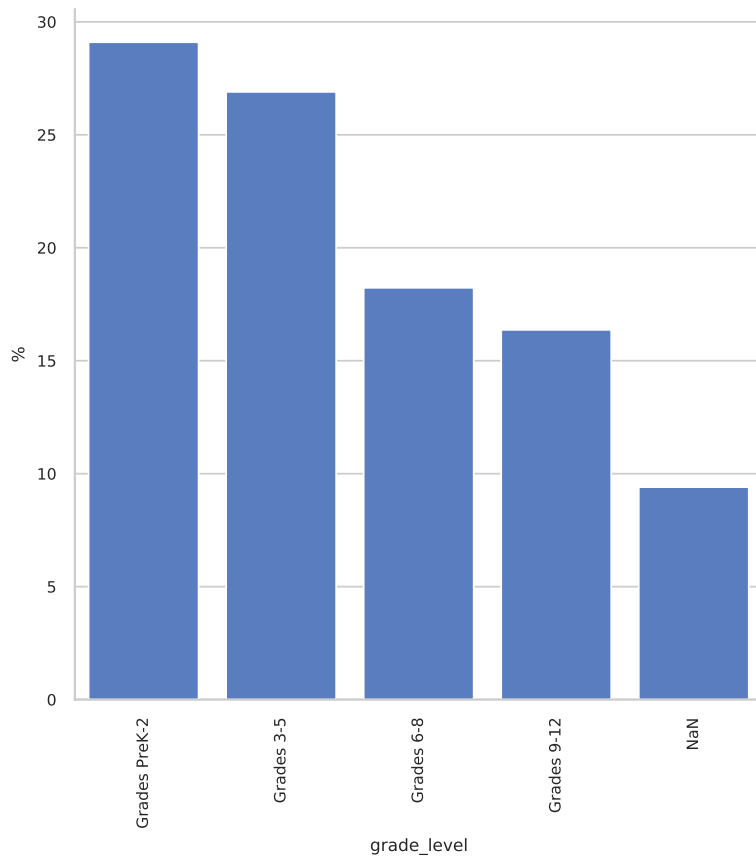


Figure A.13 : Répartition des projets par classes

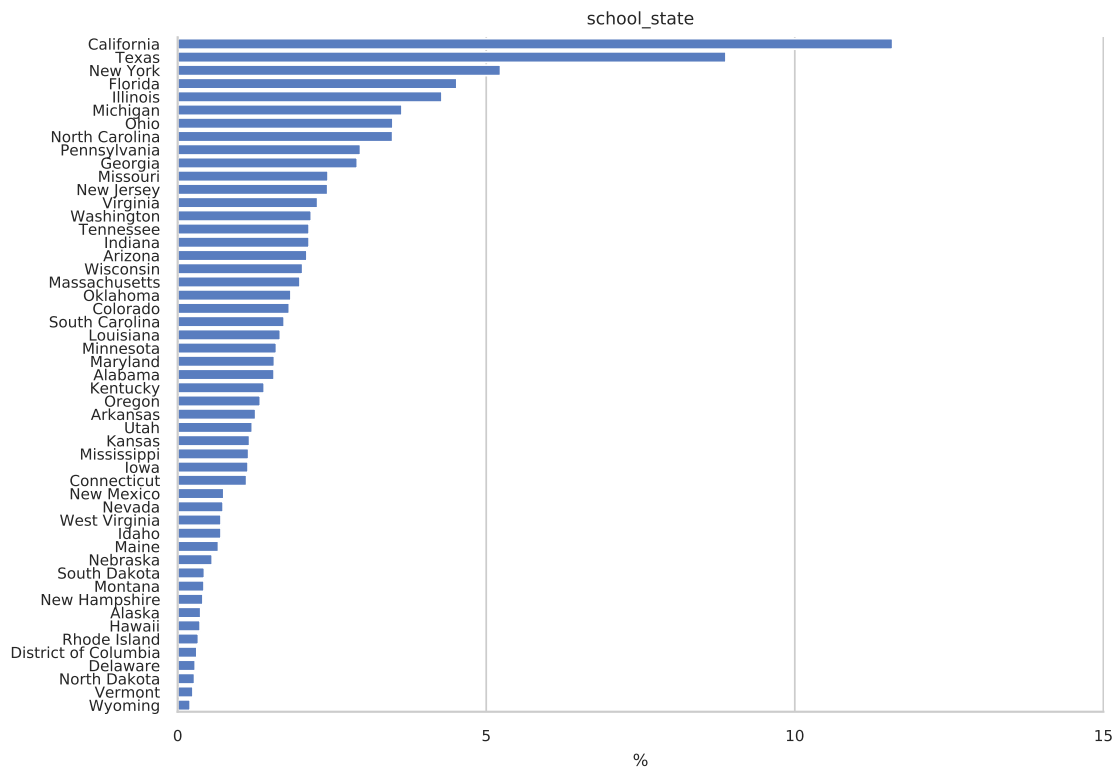


Figure A.14 : Répartition des écoles par états

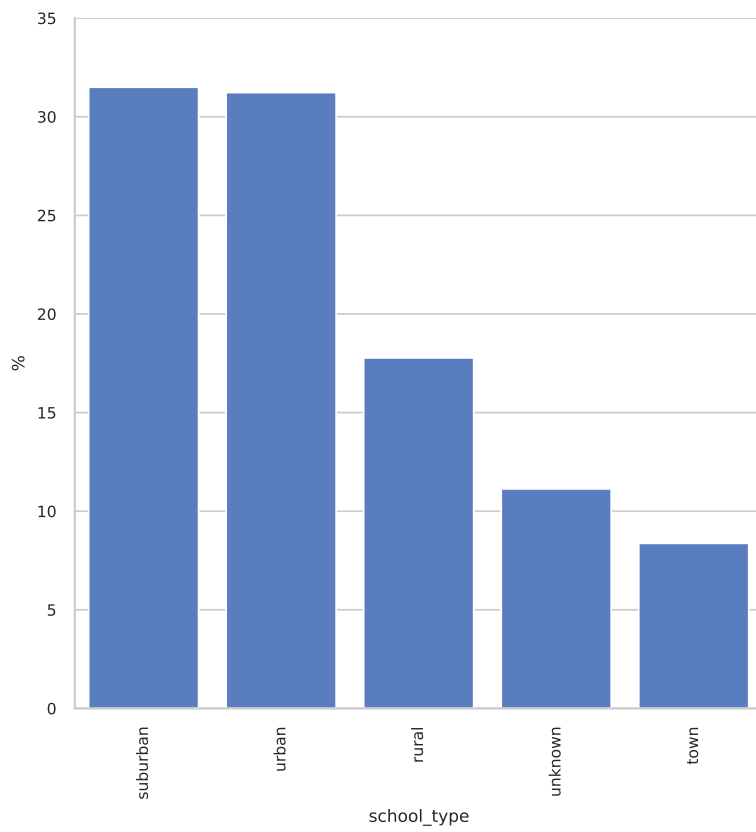


Figure A.15 : Répartition des écoles par types

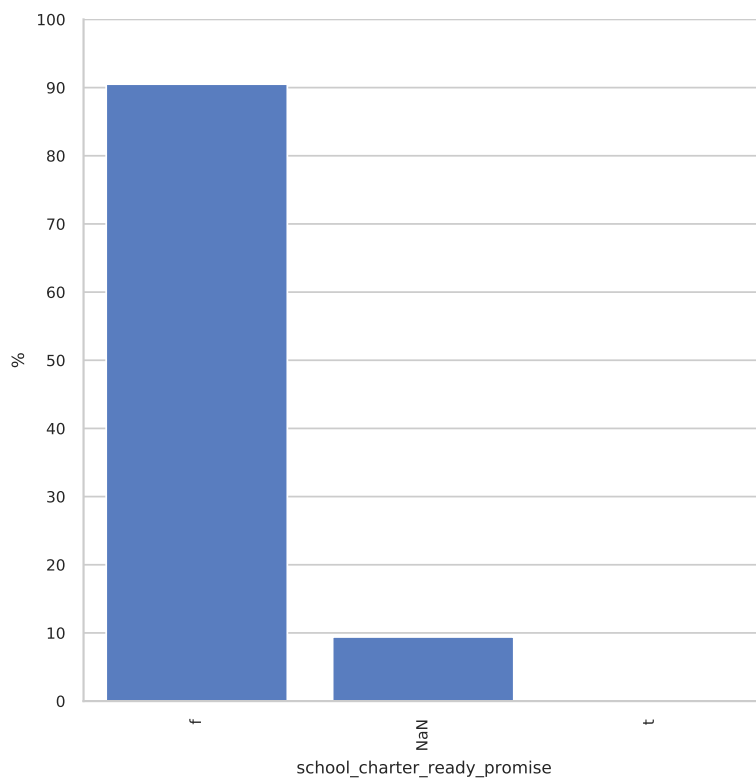


Figure A.16 : Répartition des écoles "charter ready promise"

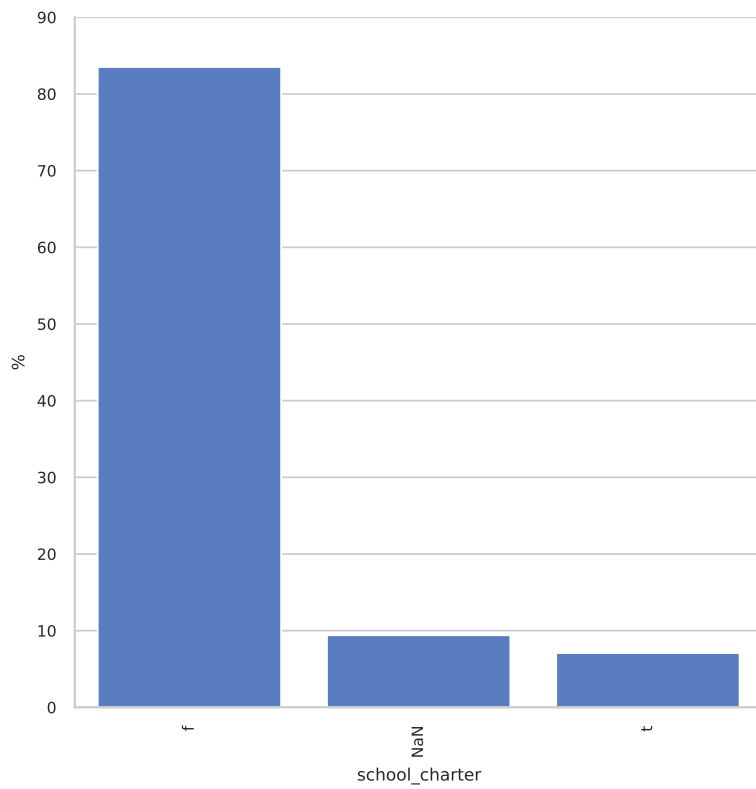


Figure A.17 : Répartition des écoles "charter"

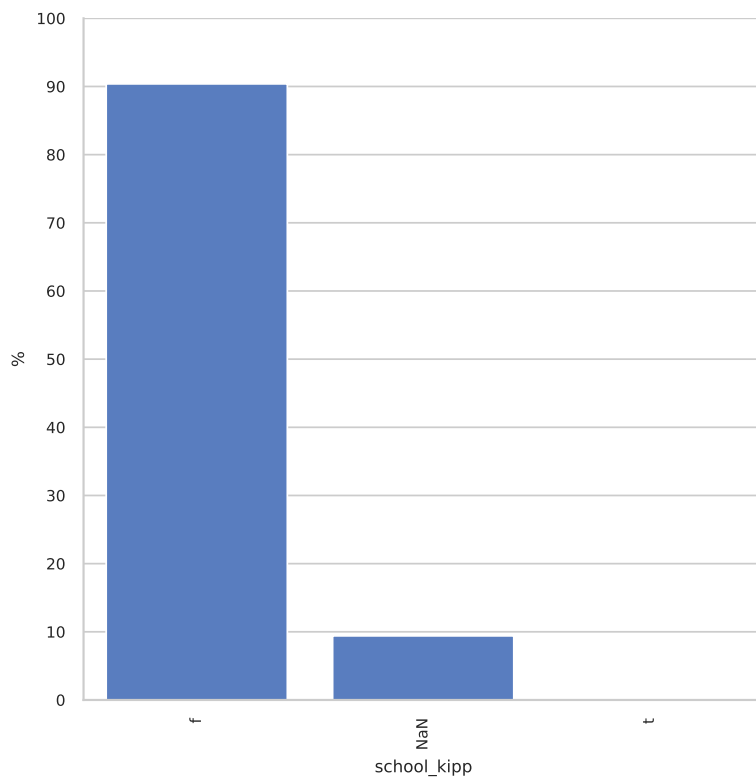


Figure A.18 : Répartition des écoles "KIPP"

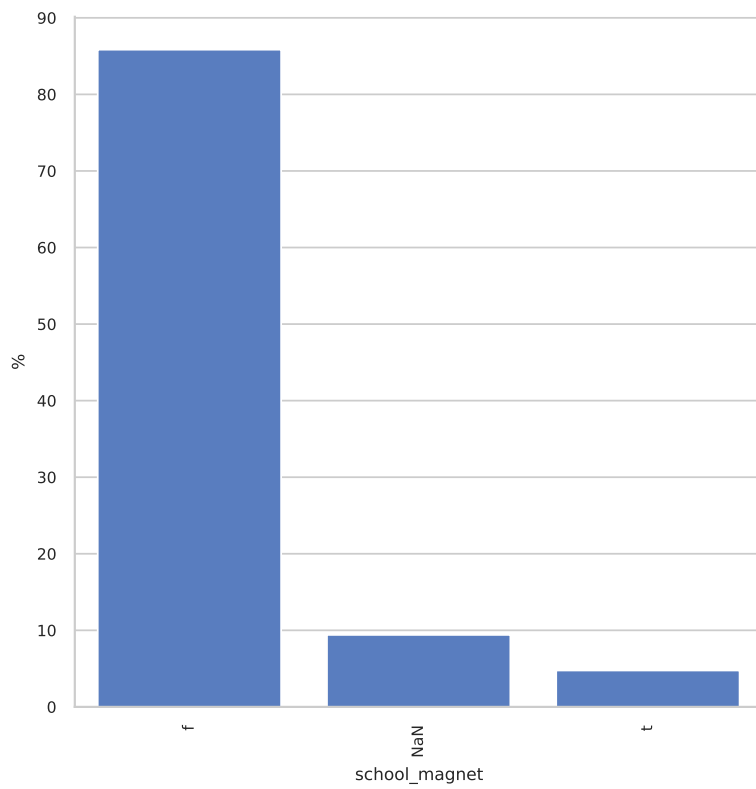


Figure A.19 : Répartition des écoles "magnet"

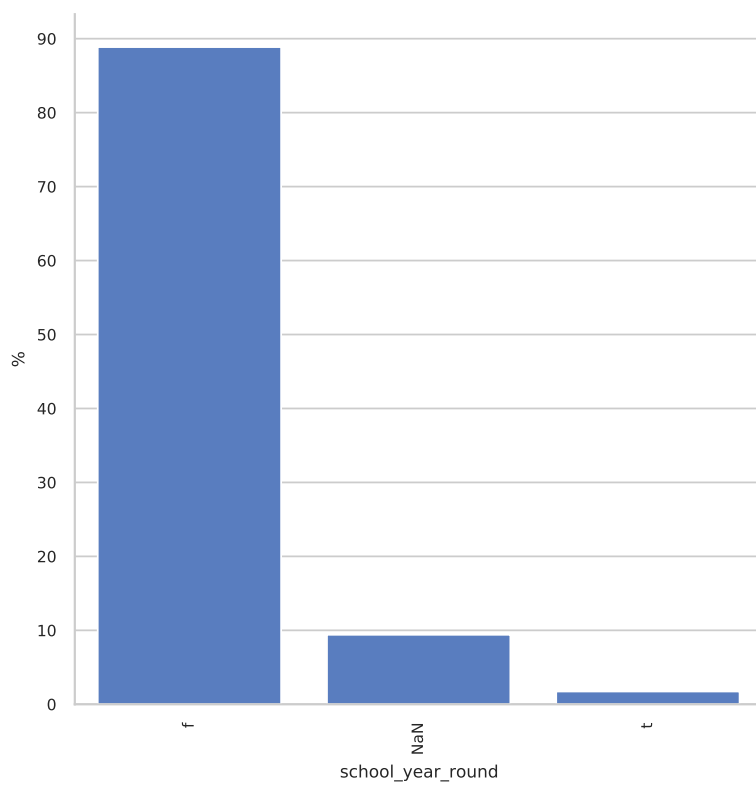


Figure A.20 : Répartition des écoles "year round"

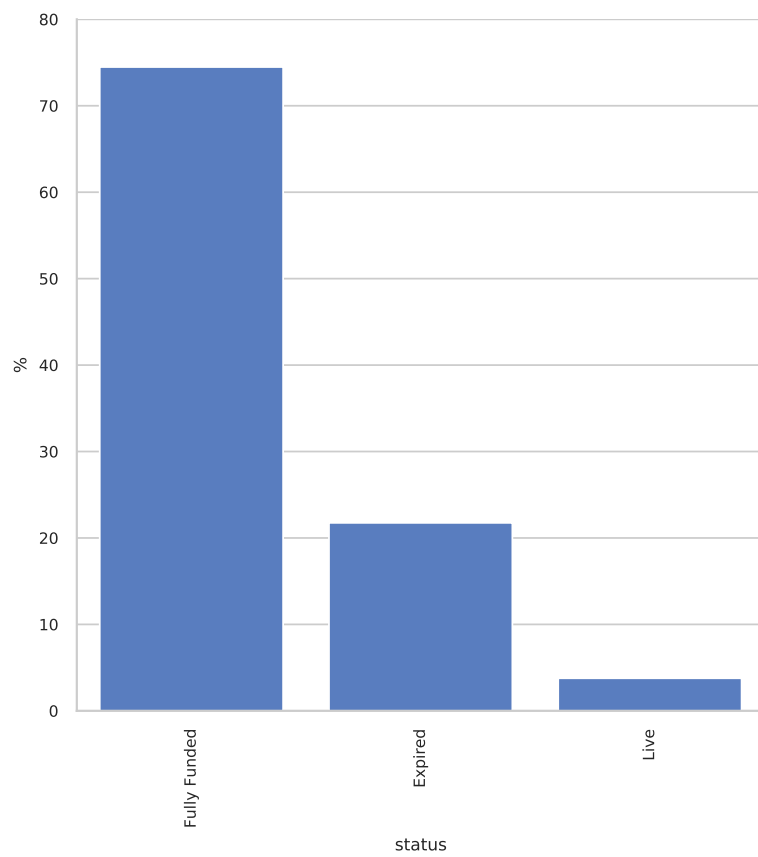


Figure A.21 : Répartition des projets par status

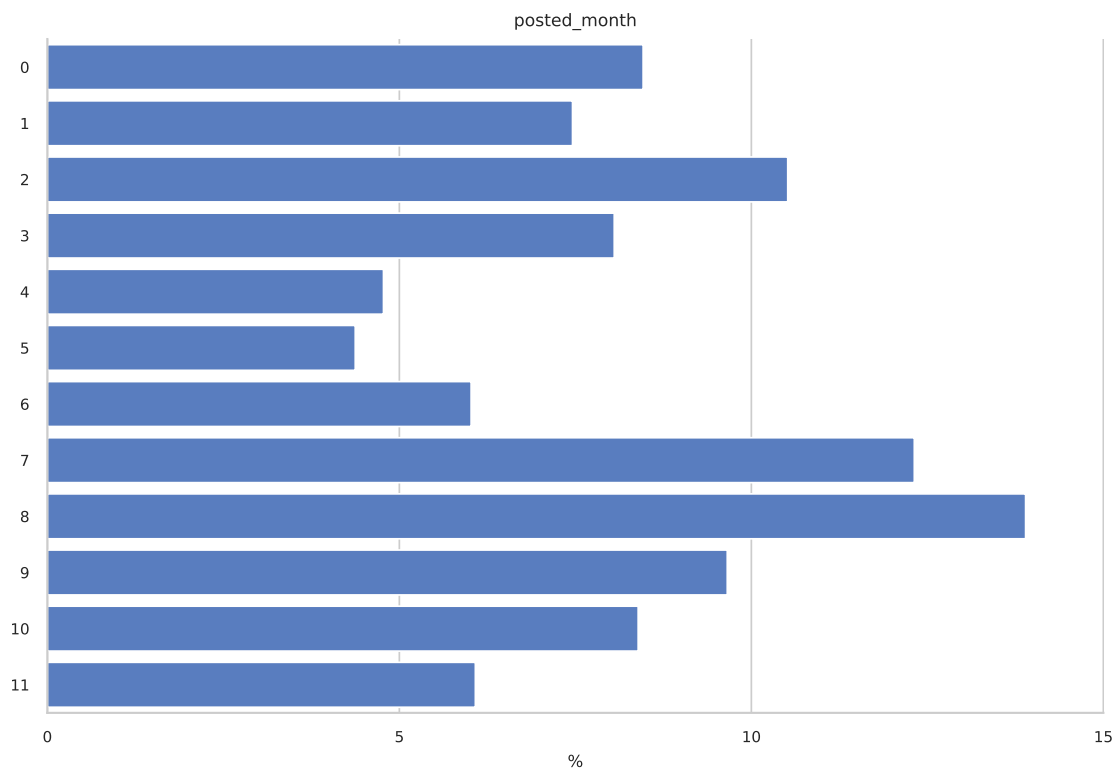


Figure A.22 : Répartition des projets par n° du mois de soumission

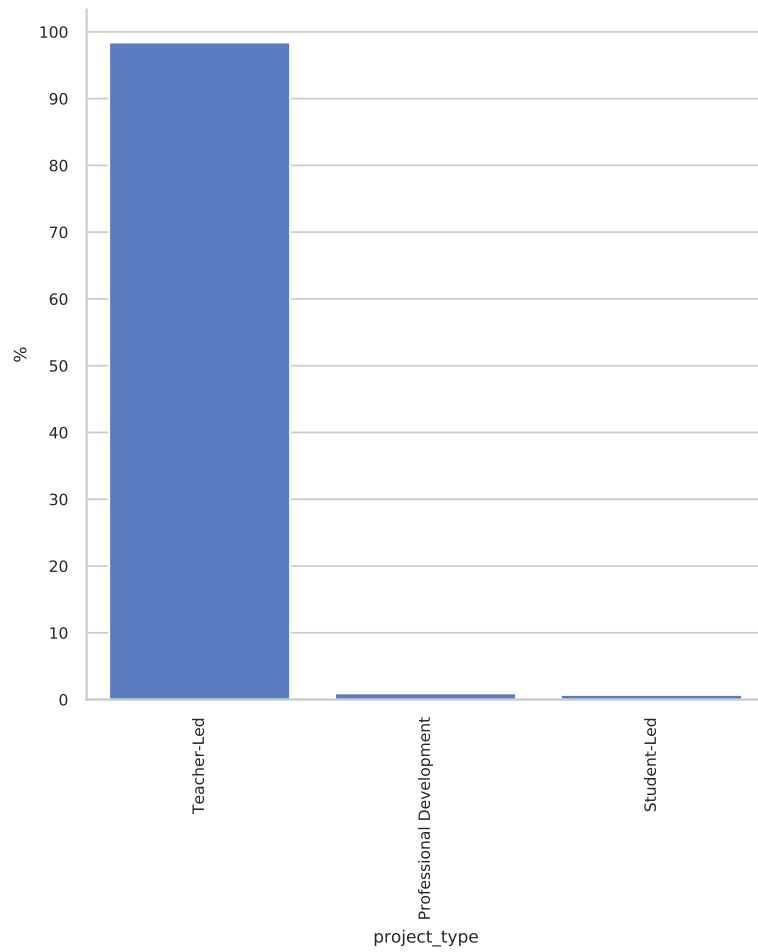


Figure A.23 : Répartition des projets par types

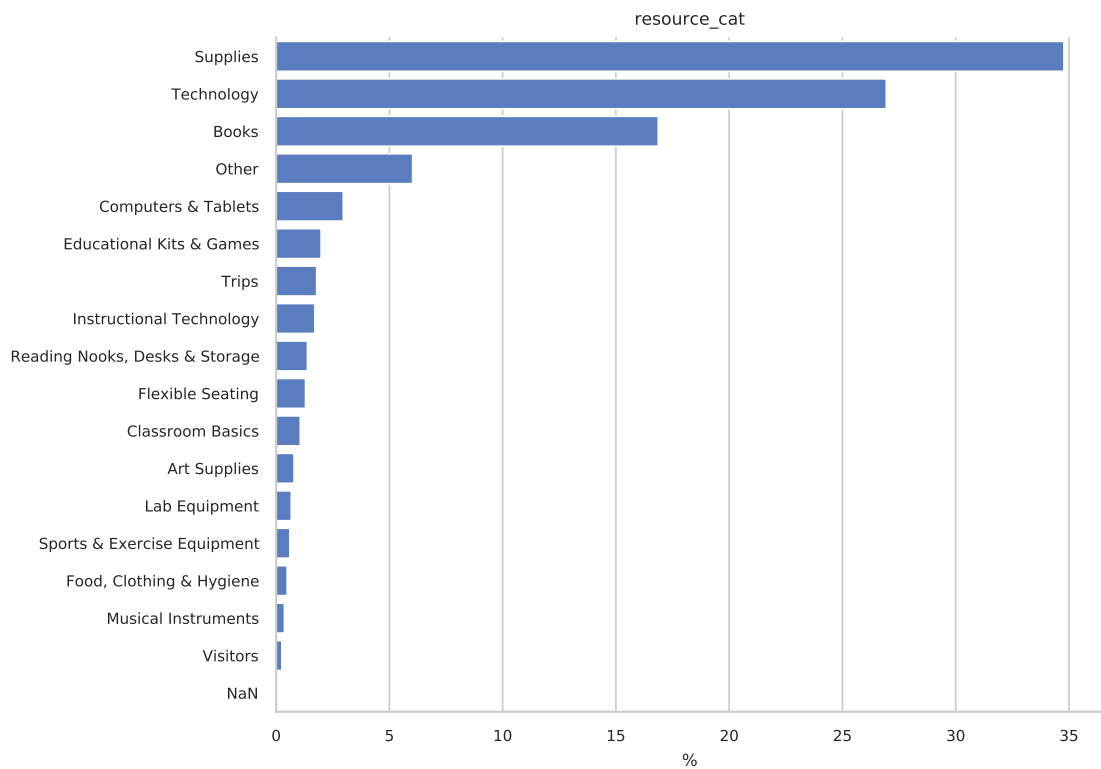


Figure A.24 : Répartition des projets par catégories de ressources

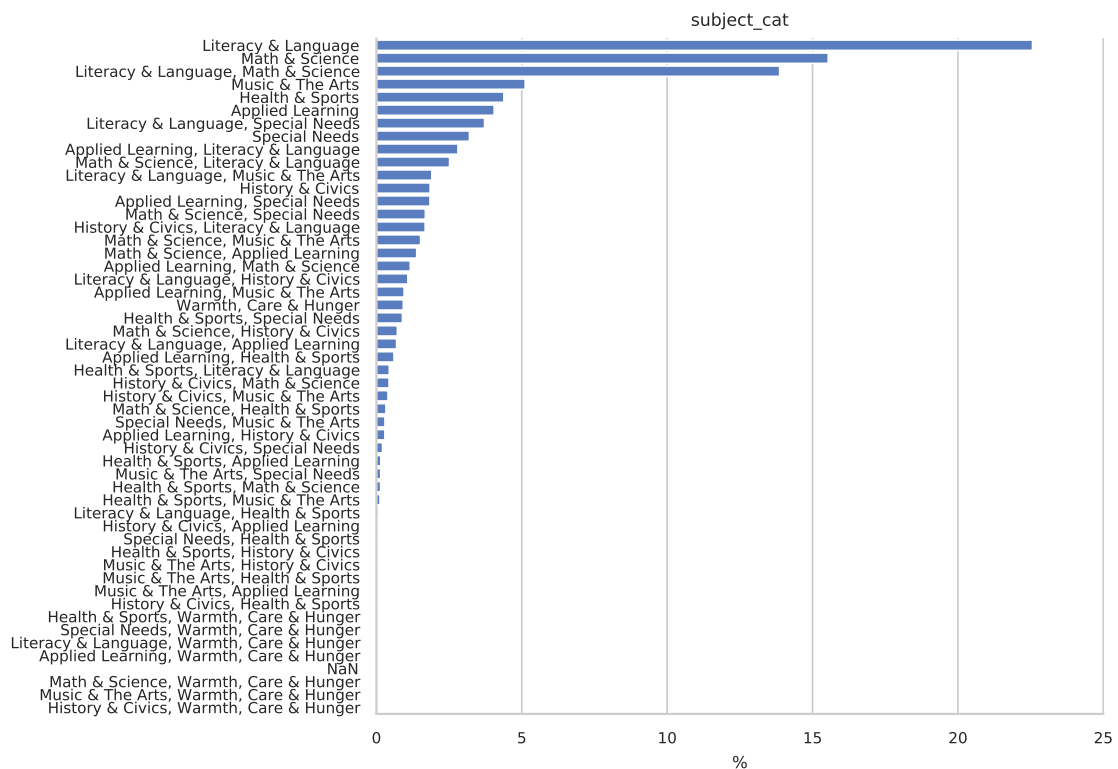


Figure A.25 : Répartition des projets par leur catégorie

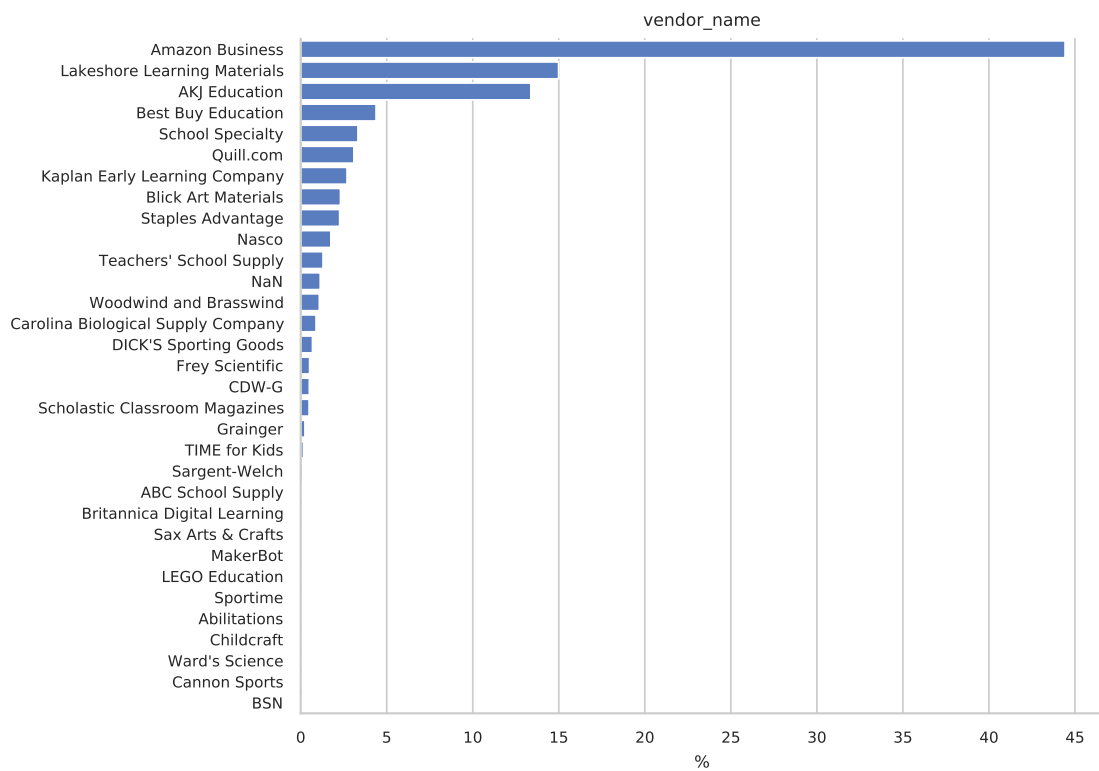


Figure A.26 : Répartition des ressources par fournisseurs

A.6 Distribution des variables quantitatives

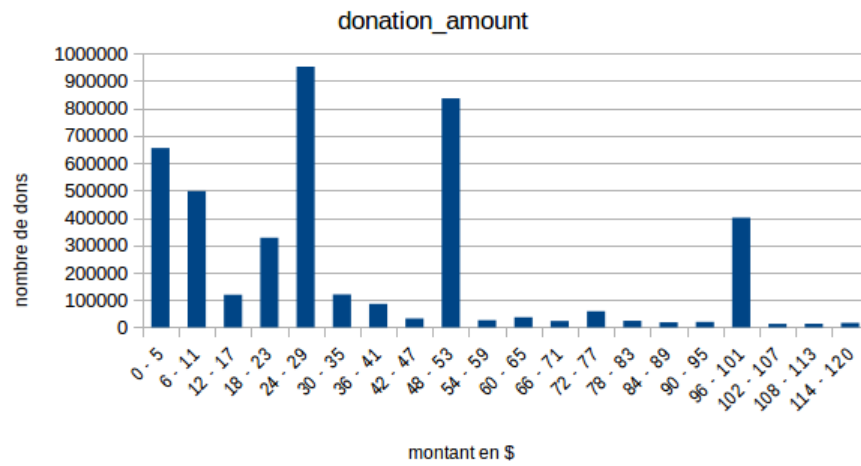


Figure A.27 : Distribution dons selon leur montant

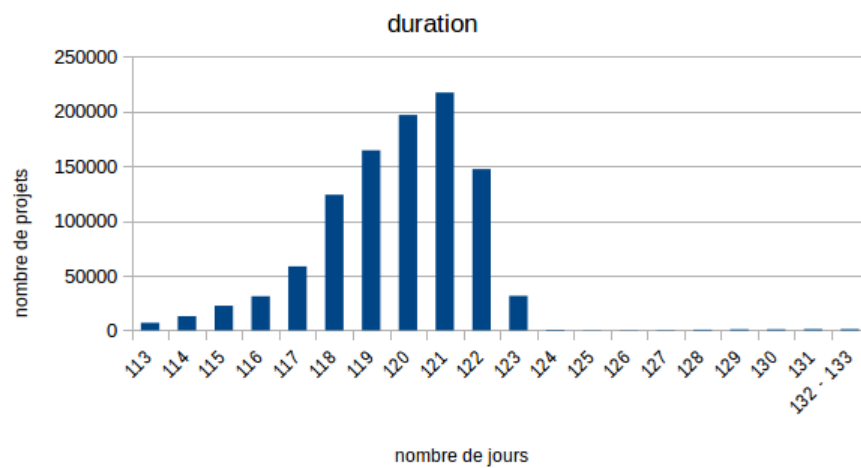


Figure A.28 : Distribution des projets selon leur durée

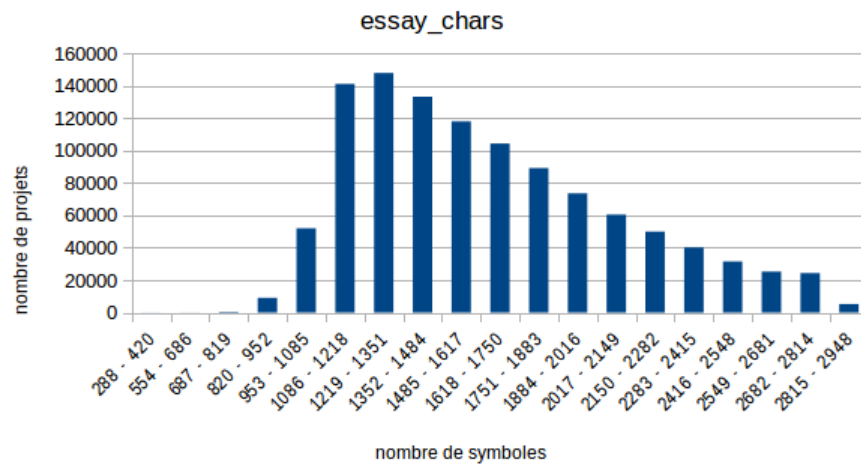


Figure A.29 : Distribution des projets selon le nombre de symboles de leur essai

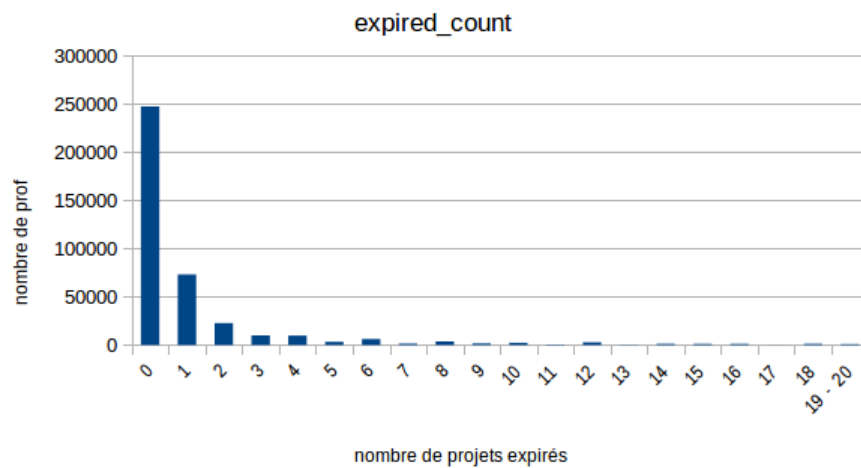


Figure A.30 : Distribution des professeurs selon leur nombre de projets échoués

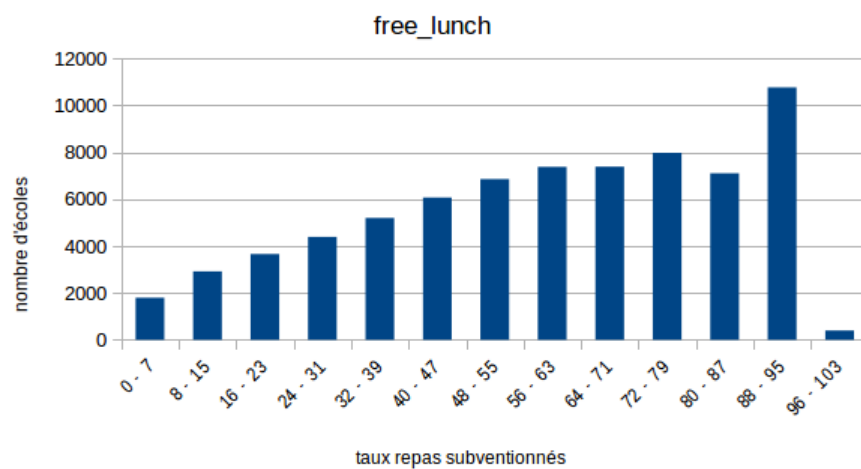


Figure A.31 : Distribution des écoles selon leur taux de repas subventionnés

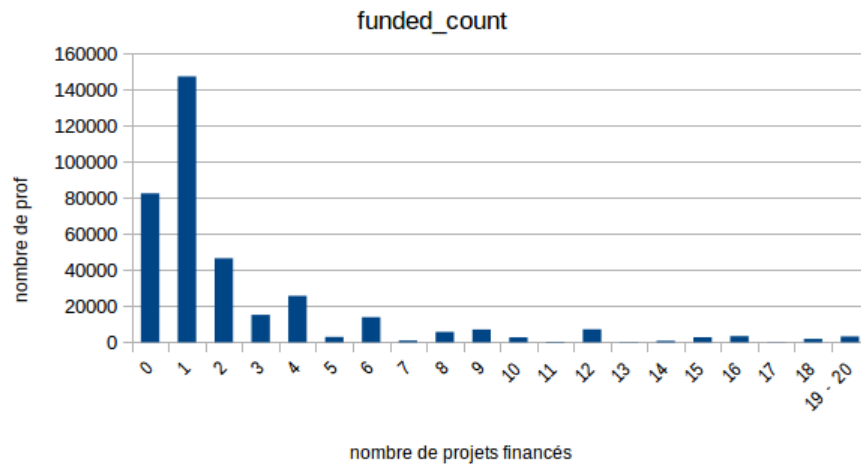


Figure A.32 : Distribution des professeurs selon leur nombre de projets financés

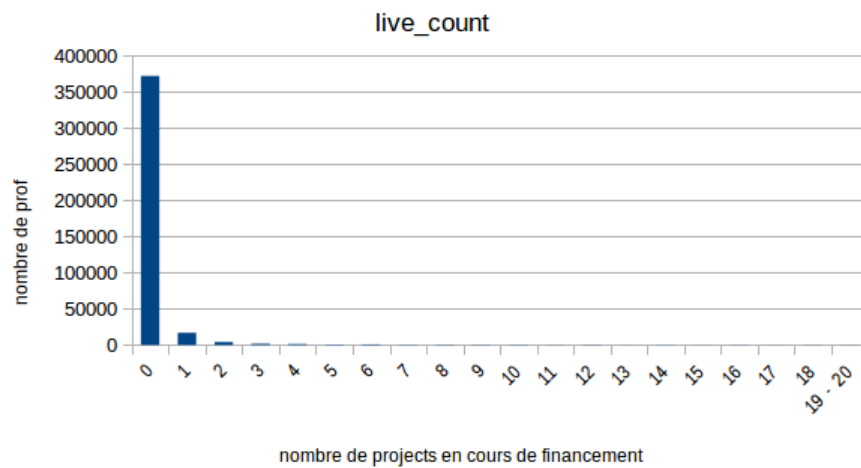


Figure A.33 : Distribution des professeurs selon leur nombre de projets actifs

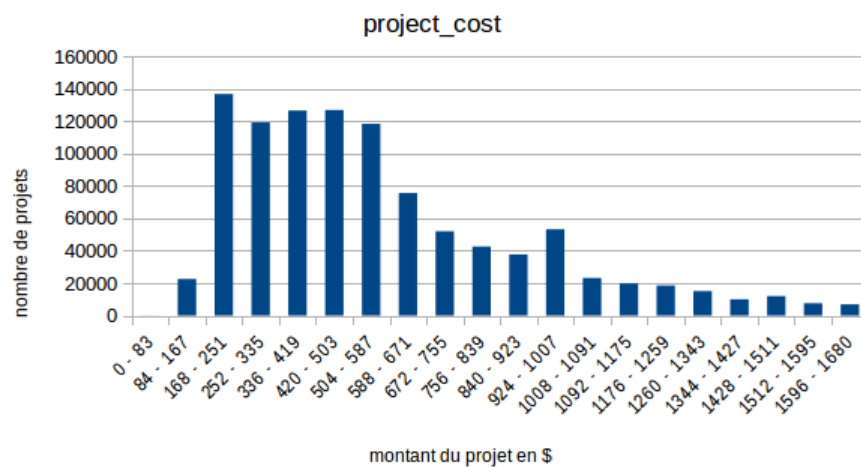


Figure A.34 : Distribution des projets selon leur coût

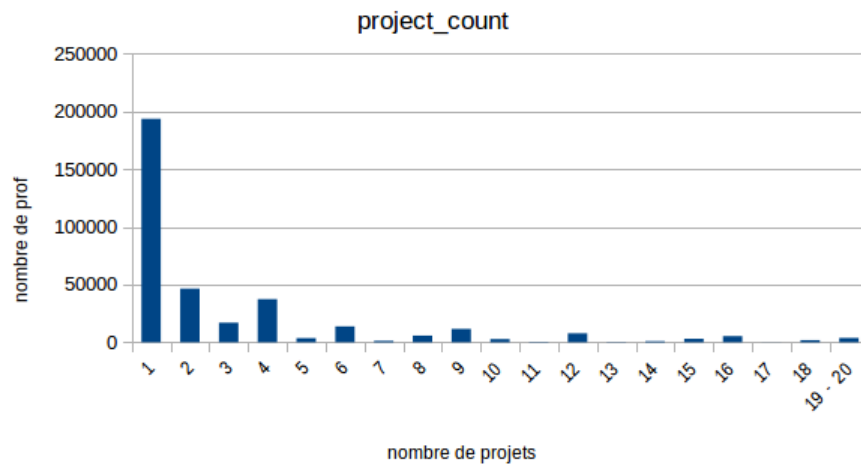


Figure A.35 : Distribution des professeurs selon leur nombre de projets soumis

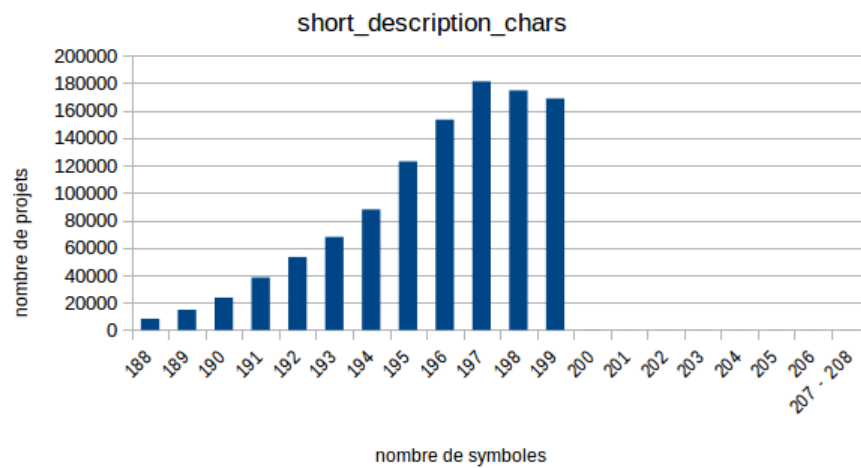


Figure A.36 : Distribution des projets selon le nombre de symboles de leur courte description

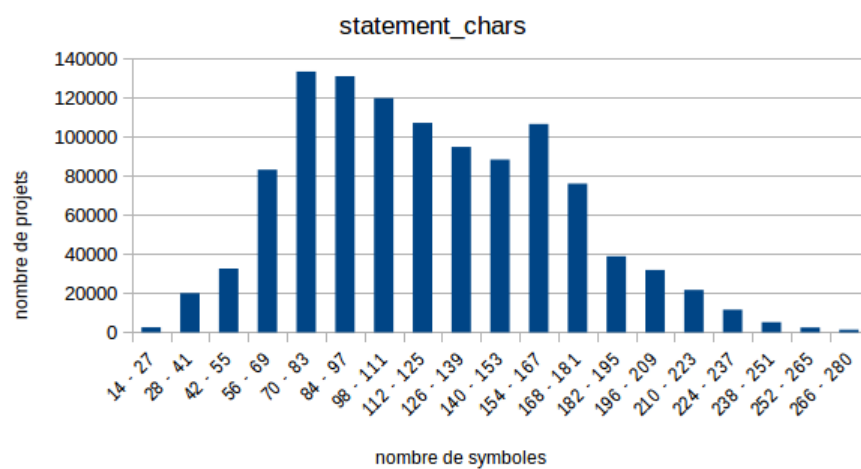


Figure A.37 : Distribution des projets selon le nombre de symboles de leur engagement

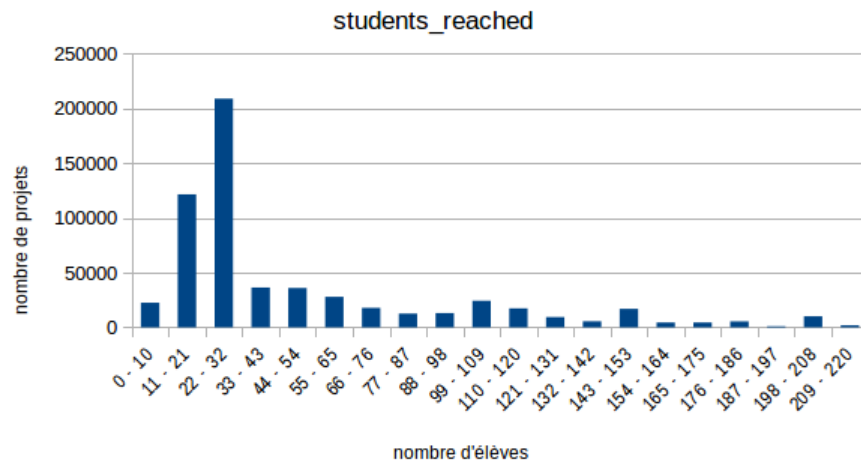


Figure A.38 : Distribution des projets selon leur nombre d'élèves impactés

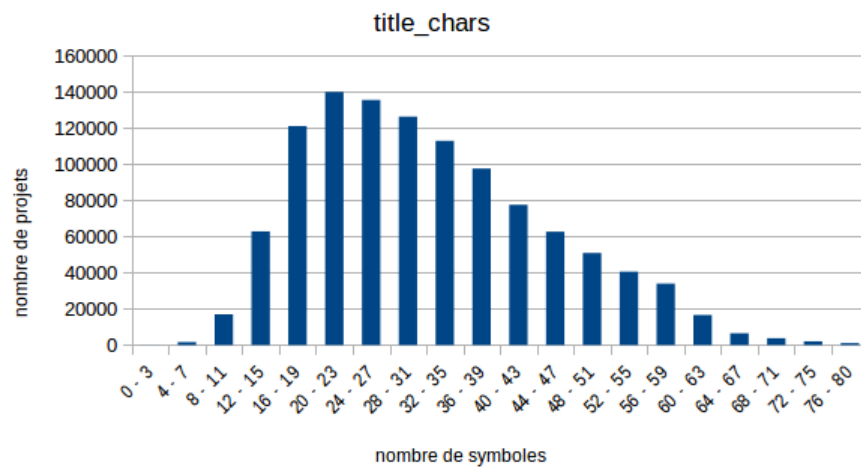


Figure A.39 : Distribution des projets selon le nombre de symboles de leur titre

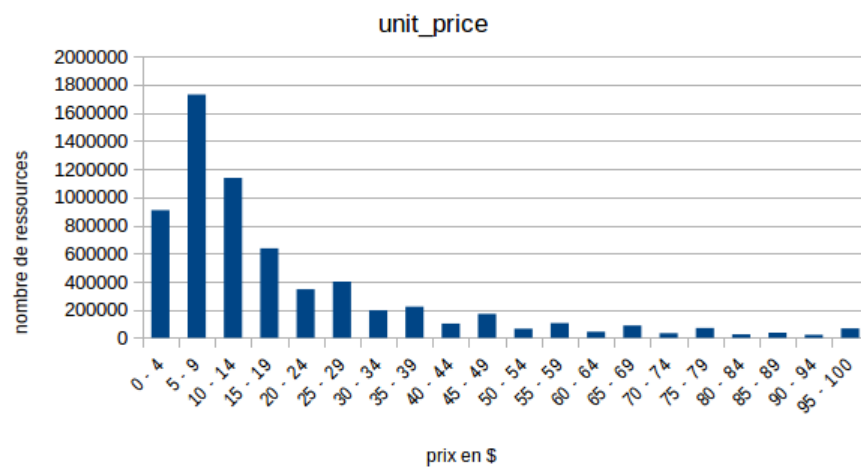


Figure A.40 : Distribution des ressources selon leur prix unitaire

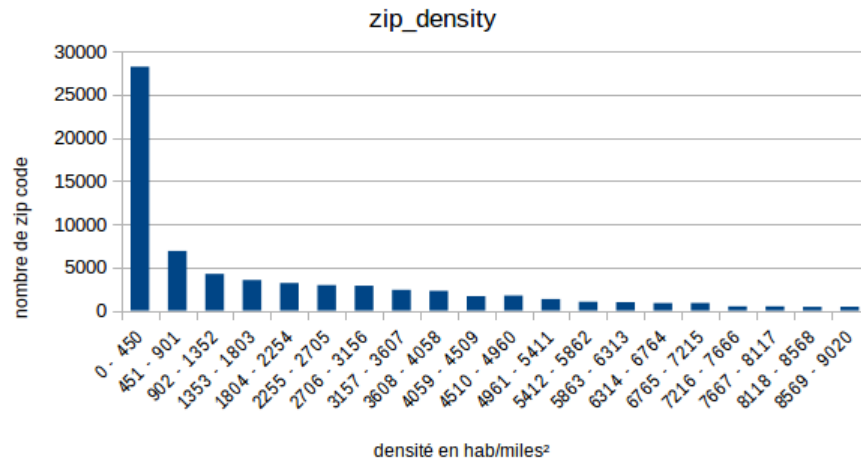


Figure A.41 : Distribution des codes postaux selon leur densité de population

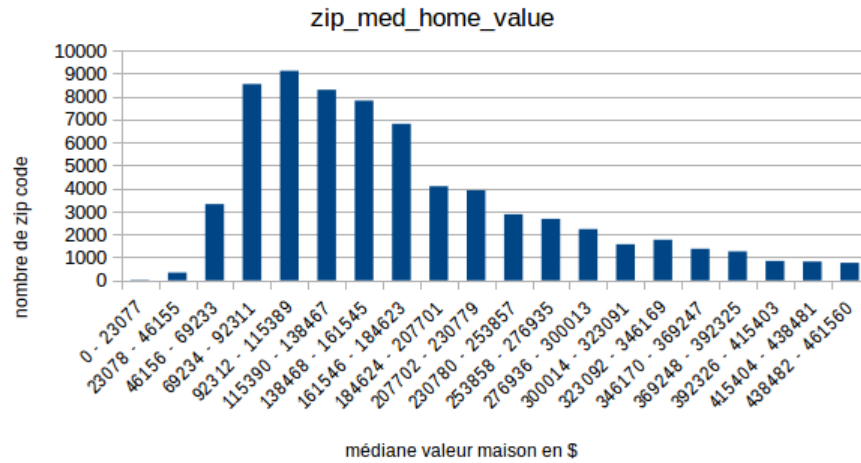


Figure A.42 : Distribution des codes postaux selon la médiane de la valeur de leurs habitations

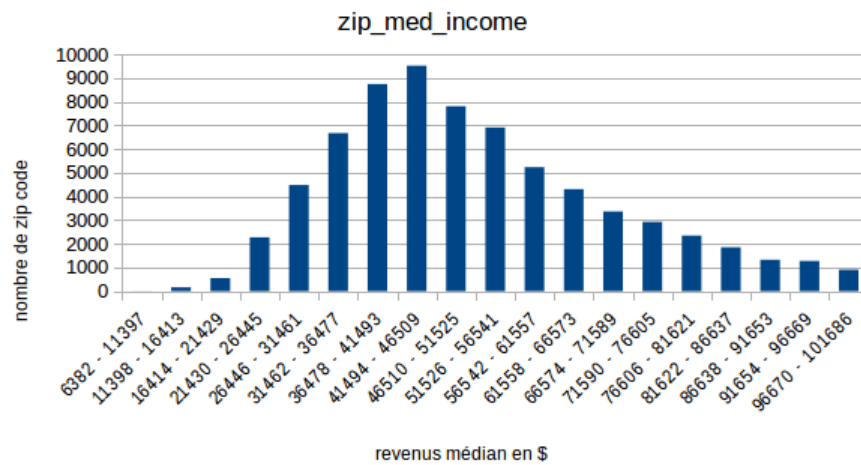


Figure A.43 : Distribution des codes postaux selon la médiane des revenus de leurs habitants

B

Corrélations

B.1 Test de Spearman

project_cost vs	corr spearman
free_lunch	-0.03
students_reached	0.07
duration	-0.03

Table B.1 : Test de Spearman project_cost

donation_amount vs	corr spearman
project_cost	0.13
students_reached	0.04
title_chars	0.00
statement_chars	0.04
essay_chars	0.00
short_description_chars	0.00
duration	-0.01

Table B.2 : Tests de Spearman donation_amount

B.2 Tests du χ^2

donor_state vs	v de Cramér
school_state	0.08

Table B.3 : Test du χ^2 / v de Cramér donor_state

poverty_level vs	v de Cramér
resource_cat	0.01
subject_cat	0.01
subject_subcat	0.00
school_type	0.08
teacher_ny_teaching_fellow	0.02
teacher_teach_for_america	0.04
school_charter	0.03
school_charter_ready_promise	0.01
school_kipp	0.02
school_magnet	0.04
school_nlns	0.03
school_year_round	0.01

Table B.4 : Test du χ^2 / v de Cramér poberty_level

status vs	v de Cramér
first_project	0.04
poverty_level	0.02
project_type	0.01
resource_cat	0.05
subject_cat	0.01
subject_subcat	0.00
posted_month	0.03
grade_level	0.01
school_type	0.03
teacher_ny_teaching_fellow	0.01
teacher_teach_for_america	0.04
school_charter	0.01
school_charter_ready_promise	0.00
school_kipp	0.01
school_magnet	0.01
school_nlns	0.01
school_year_round	0.01
school_county	0.00
fixed_teacher_prefix	0.00

Table B.5 : Test du χ^2 / v de Cramér status

B.3 Régressions Logistiques

donation_amount vs	métrique	score
donor_state	f1	0.04
is_teacher	auc	0.68
month	f1	0.03
status	auc	0.51
grade_level	f1	0.21
resource_cat	f1	0.20
first_project	auc	0.57
posted_month	f1	0.05
subject_cat	f1	0.08

Table B.6 : Régression logistique donation_amount

poverty_level vs	métrique	score
zip_med_income	f1	0.55
zip_density	f1	0.42
zip_med_home_value	f1	0.43
free_lunch	f1	0.86

Table B.7 : Régression logistique poverty_level

project_cost vs	métrique	score
first_project	auc	0.59
resource_cat	f1	0.30
subject_cat	f1	0.11
posted_month	f1	0.06
grade_level	f1	0.24
poverty_level	f1	0.42
school_type	f1	0.29
teacher_ny_teaching_fellow	auc	0.51
teacher_teach_for_america	auc	0.51
school_charter	auc	0.52
school_charter_ready_promise	auc	0.53
school_kipp	auc	0.56
school_magnet	auc	0.50
school_nlns	auc	0.53
school_year_round	auc	0.52

Table B.8 : Régression logistique project_cost

status vs	métrique	score
free_lunch	auc	0.53
zip_med_income	auc	0.51
zip_density	auc	0.54
zip_med_home_value	auc	0.52
title_chars	auc	0.50
statement_chars	auc	0.50
essay_chars	auc	0.50
short_description_chars	auc	0.50
project_cost	auc	0.67
students_reached	auc	0.51
duration	auc	0.51

Table B.9 : Régression logistique status

C

Configurations

C.1 Spécifications matérielles

Le cluster est composé de trois nœuds :

- 1 x HP Z400, CPU : Intel Xeon CPU W3520 2.67GHz (4 cœurs), 16 Go RAM
- 2 x Dell Optiplex, CPU : Intel Core i5 2.90GHz (4 cœurs), 16 Go RAM

C.2 Spécifications logicielles

- Spark 2.3.1
- Scala 2.11.8
- Java 1.8.0_151
- Python 3.7
- MongoDB 3.6.5
- Docker 18.06.1-ce
- CentOS 7.5 et Linux Mint 18.2

D

Modélisation

D.1 AFDM

Calcul du facteur à appliquer sur les variables binaires provenant du codage one-hot des variables qualitatives :

Soit n_k l'effectif d'une modalité d'une variable binaire et n le nombre d'enregistrements :

$$\text{proportion} = \frac{n_k}{n}$$

$$\text{facteur} = \frac{1}{\sqrt{\text{proportion}}}$$

D.2 Arbre de décisions

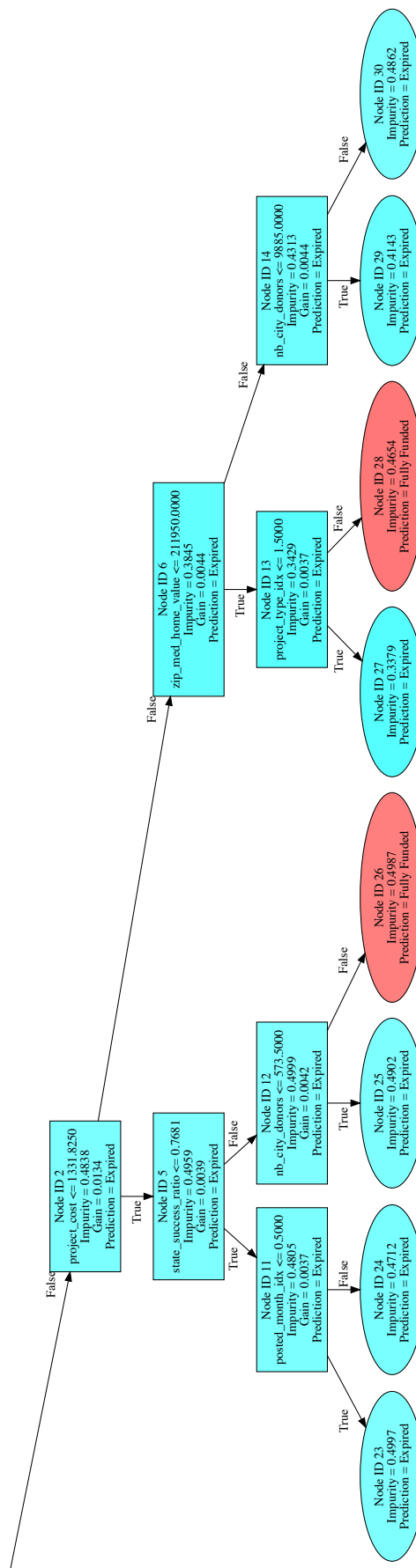


Figure D.1 : Exemple d'arbre de décision (profondeur 4) 1/2

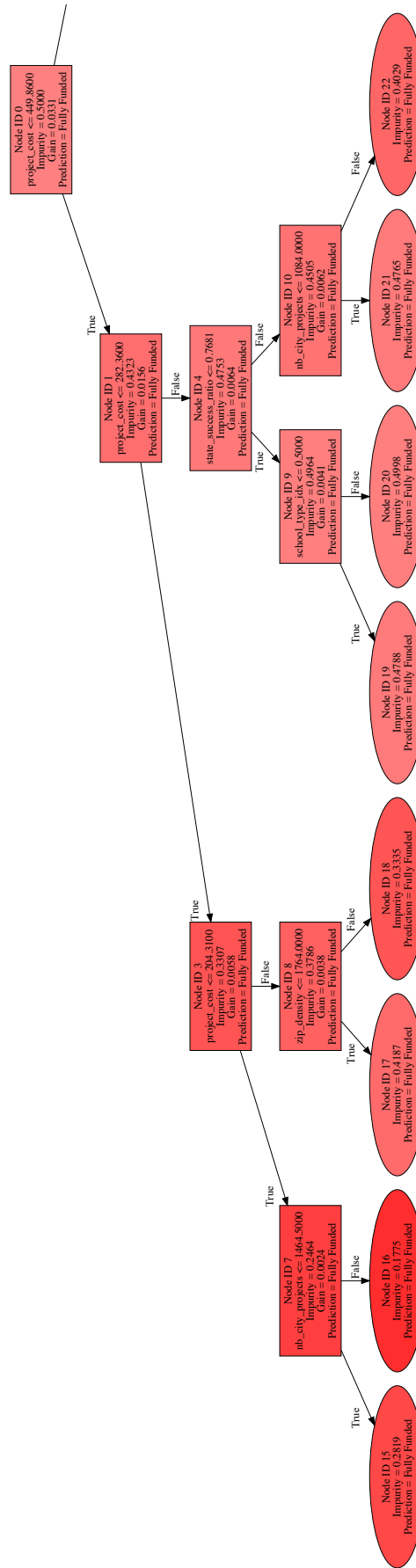


Figure D.2 : Exemple d'arbre de décision (profondeur 4) 2/2

Bibliographie

- [1] Advanced Analytics. *Using Scala UDFs in PySpark*. 2018. url : <https://medium.com/wbaa/using-scala-udfs-in-pyspark-b70033dd69b9> (visité le 16/09/2018).
- [2] David M. Blei, Andrew Y. Ng et Michael I. Jordan. “Latent Dirichlet Allocation”. In : *J. Mach. Learn. Res.* 3 (mar. 2003), p. 993–1022. issn : 1532-4435. url : <http://dl.acm.org/citation.cfm?id=944919.944937>.
- [3] Leo Breiman. “Random Forests”. In : *Machine Learning* 45.1 (oct. 2001), p. 5–32. issn : 1573-0565. doi : [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324). url : <https://doi.org/10.1023/A:1010933404324>.
- [4] Donors Choose. *Download data*. 2018. url : <https://research.donorschoose.org/t/download-opendata/33> (visité le 16/09/2018).
- [5] Donors Choose. *Help DonorsChoose.org connect donors with projects they care about*. 2018. url : <https://www.kaggle.com/donorschoose/io/home> (visité le 16/09/2018).
- [6] Donors Choose. *Improve education with data*. 2018. url : <https://data.donorschoose.org/> (visité le 16/09/2018).
- [7] Donors Choose. *Opedata Layout and Docs*. 2018. url : <https://research.donorschoose.org/t/opedata-layout-and-docs/18> (visité le 16/09/2018).
- [8] Donors Choose. *Predict whether teachers’ project proposals are accepted*. 2018. url : <https://www.kaggle.com/c/donorschoose-application-screening> (visité le 16/09/2018).
- [9] Donors Choose. *Schools We Have Yet to Reach*. 2018. url : <https://data.donorschoose.org/no-school-left-behind-get-your-school-on-the-map/> (visité le 16/09/2018).
- [10] Donors Choose. *Support a classroom. Build a future*. 2018. url : <https://www.donorschoose.org/> (visité le 16/09/2018).
- [11] Donors Choose. *What teachers need*. 2018. url : <https://data.donorschoose.org/apps/tech-in-classrooms/#risingDemand> (visité le 16/09/2018).
- [12] Stanford CoreNLP. *Stanford CoreNLP – natural language software*. 2018. url : <https://stanfordnlp.github.io/CoreNLP/> (visité le 16/09/2018).

- [13] Flare : *Optimizing Apache Spark for Scale-Up Architectures and Medium-Size Data*. OSDI. 2018. url : <https://www.cs.purdue.edu/homes/rompf/papers/essertel-osdi18.pdf>.
- [14] Gate. *General architecture for text engineering*. 2018. url : <https://gate.ac.uk/> (visité le 16/09/2018).
- [15] Graphviz. *Graphviz - Graph Visualization Software*. 2018. url : <https://www.graphviz.org/> (visité le 16/09/2018).
- [16] Karen Sparck Jones. “A statistical interpretation of term specificity and its application in retrieval”. In : *Journal of documentation* 28.1 (1972), p. 11–21. doi : 10.1108/eb026526. eprint : <https://doi.org/10.1108/eb026526>. url : <https://doi.org/10.1108/eb026526>.
- [17] Armand Joulin et al. “Bag of Tricks for Efficient Text Classification”. In : *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*. Association for Computational Linguistics, avr. 2017, p. 427–431.
- [18] Kickstarter. *Kickstarter*. 2018. url : <https://www.kickstarter.com> (visité le 16/09/2018).
- [19] M.A. Kuijer et E. Haasdijk. “Predicting school funding requests that deserve an A+”. Mém.de mast. Universiteit Amsterdam, 2014. url : https://beta.vu.nl/nl/Images/werkstuk-Kuijer_tcm235-413416.pdf.
- [20] John Snow Labs. *High Performance NLP with Apache Spark*. 2018. url : <https://nlp.johnsnowlabs.com/> (visité le 16/09/2018).
- [21] Mallet. *MAchine Learning for LanguagE Toolkit*. 2018. url : <http://mallet.cs.umass.edu/> (visité le 16/09/2018).
- [22] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In : *CoRR abs/1301.3781* (2013). arXiv : 1301.3781. url : <http://arxiv.org/abs/1301.3781>.
- [23] MongoDB. *BSON Types*. 2018. url : <https://docs.mongodb.com/manual/reference/bson-types/> (visité le 16/09/2018).
- [24] MySQL. *Full-Text Stopwords*. 2018. url : <https://dev.mysql.com/doc/refman/5.5/en/fulltext-stopwords.html> (visité le 16/09/2018).
- [25] OpenNLP. *OpenNLP*. 2018. url : <https://opennlp.apache.org/> (visité le 16/09/2018).
- [26] *Proposing a Recommendation system for DonorsChoose.org*. SAS Global Forum. 2016. url : <https://support.sas.com/resources/papers/proceedings16/12526-2016.pdf>.
- [27] Rico Rakotomalala. *Analyse factorielle de données mixtes*. 2018. url : <http://eric.univ-lyon2.fr/%7Ericco/cours/slides/AFDM.pdf> (visité le 16/09/2018).
- [28] Ranks. *Stopword lists*. 2018. url : <https://www.ranks.nl/stopwords> (visité le 16/09/2018).
- [29] Facebook research. *Library for fast text representation and classification*. 2018. url : https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md?utm_campaign=buffer&utm_content=buffer0df9b&utm_medium=social&utm_source=linkedin.com (visité le 16/09/2018).
- [30] Harlan Seymour. *1st place solution*. 2018. url : <https://www.kaggle.com/shadowwarrior/1st-place-solution> (visité le 16/09/2018).
- [31] Julio Antonio Soto. *spark-tree-plotting*. 2018. url : <https://github.com/julioasotodv/spark-tree-plotting> (visité le 16/09/2018).

-
- [32] spaCy. *Industrial-strength natural language processing in Python*. 2018. url : <https://spacy.io/> (visité le 16/09/2018).
 - [33] Spark. *Word2Vec import/export for original binary format*. 2018. url : <https://issues.apache.org/jira/browse/SPARK-9484> (visité le 16/09/2018).
 - [34] Textfixer. *common english words*. 2018. url : <https://www.textfixer.com/tutorials/common-english-words.txt> (visité le 16/09/2018).
 - [35] Textfixer. *Most Common Words in English*. 2018. url : <https://www.textfixer.com/tutorials/common-english-words.php> (visité le 16/09/2018).
 - [36] Apache UIMA. *Apache UIMA*. 2018. url : <https://uima.apache.org/> (visité le 16/09/2018).
 - [37] Ulule. *Donnez vie aux bonnes idées*. 2018. url : <https://fr.ulule.com> (visité le 16/09/2018).
 - [38] UnitedStatesZipCodes.org. *United States Zip Codes*. 2018. url : <https://www.unitedstateszipcodes.org/> (visité le 16/09/2018).
 - [39] wikipedia. *Contraction*. 2018. url : https://en.wikipedia.org/wiki/Contraction_%28grammar%29#English (visité le 16/09/2018).
 - [40] XPO6. *List of english stop words*. 2018. url : <http://xpo6.com/list-of-english-stop-words/> (visité le 16/09/2018).

Table des figures

1.1	Processus d'exploration et de modélisation	3
2.1	Architecture du cluster	6
2.2	Architecture scalable	8
3.1	Diagramme MCD du dataset Kaggle	12
4.1	Pipeline diversité lexicale	14
4.2	Pipeline lemmatisation	15
5.1	Optimisation de la profondeur d'arbre (RF)	23
5.2	Optimisation du nombre d'arbres (RF)	23
5.3	Importance des 12 premières variables (RF)	24
5.4	Optimisation du nombre d'itérations (GBT)	25
5.5	Importance des 12 premières variables (GBT)	25
5.6	Score AUC pour les optimums trouvés	26
A.1	Description d'un projet	30
A.2	Détails sur le coût du projet	31
A.3	Activités autour du projet	32
A.4	Répartition des donateurs par états	41
A.5	Répartition des dons par n° de mois	42
A.6	Répartition du don optionnel	42
A.7	Répartition des professeurs parmi les donateurs	43
A.8	Répartition des premiers projets	43
A.9	Répartition des professeurs "ny teaching fellow"	44
A.10	Répartition des professeurs par leur préfixes	44
A.11	Répartition des professeurs "teach for america"	45
A.12	Répartition des écoles par niveaux de pauvreté	46
A.13	Répartition des projets par classes	47
A.14	Répartition des écoles par états	47
A.15	Répartition des écoles par types	48
A.16	Répartition des écoles "charter ready promise"	48
A.17	Répartition des écoles "charter"	49

A.18 Répartition des écoles "KIPP"	49
A.19 Répartition des écoles "magnet"	50
A.20 Répartition des écoles "year round"	50
A.21 Répartition des projets par status	51
A.22 Répartition des projets par n° du mois de soumission	51
A.23 Répartition des projets par types	52
A.24 Répartition des projets par catégories de ressources	53
A.25 Répartition des projets par leur catégorie	53
A.26 Répartition des ressources par fournisseurs	54
A.27 Distribution dons selon leur montant	55
A.28 Distribution des projets selon leur durée	55
A.29 Distribution des projets selon le nombre de symboles de leur essai	56
A.30 Distribution des professeurs selon leur nombre de projets échoués	56
A.31 Distribution des écoles selon leur taux de repas subventionnés	56
A.32 Distribution des professeurs selon leur nombre de projets financés	57
A.33 Distribution des professeurs selon leur nombre de projets actifs	57
A.34 Distribution des projets selon leur coût	57
A.35 Distribution des professeurs selon leur nombre de projets soumis	58
A.36 Distribution des projets selon le nombre de symboles de leur courte description	58
A.37 Distribution des projets selon le nombre de symboles de leur engagement	58
A.38 Distribution des projets selon leur nombre d'élèves impactés	59
A.39 Distribution des projets selon le nombre de symboles de leur titre	59
A.40 Distribution des ressources selon leur prix unitaire	59
A.41 Distribution des codes postaux selon leur densité de population	60
A.42 Distribution des codes postaux selon la médiane de la valeur de leurs habitations	60
A.43 Distribution des codes postaux selon la médiane des revenus de leurs habitants	60
D.1 Exemple d'arbre de décision (profondeur 4) 1/2	67
D.2 Exemple d'arbre de décision (profondeur 4) 2/2	68

Liste des tableaux

2.1	Description des tables du dataset Kaggle	5
4.1	Sélection des mots TF-IDF	17
5.1	Compositions des datasets pour la modélisation	19
5.2	AFDM : pourcentages de variance expliquée pour les premiers axes factoriels	21
5.3	Configuration de l'arbre de décision	21
5.4	Hyperparamètres spécifiques à Random Forest	22
5.5	Configuration de l'optimisation	22
5.6	Hyperparamètres Gradient Boosting	24
A.1	Variables de la table schools	33
A.2	Variables de la table projects	34
A.3	Variables de la table donations	35
A.4	Variables de la table resources	35
A.5	Variables de la table teachers	35
A.6	Variables de la table donors	35
A.7	Variables quantitatives de la table projects 1/3	36
A.8	Variables quantitatives de la table projects 2/3	36
A.9	Variables quantitatives de la table projects 3/3	37
A.10	Variables quantitatives de la table donations	37
A.11	Variables quantitatives de la table resources	37
A.12	Variables quantitatives de la table schools	38
A.13	Variables quantitatives de la table teachers 1/2	38
A.14	Variables quantitatives de la table teachers 2/2	38
A.15	Variables qualitatives de la table projects	39
A.16	Variables qualitatives de la table donations	39
A.17	Variables qualitatives de la table resources	39
A.18	Variables qualitatives de la table schools	40
A.19	Variables qualitatives de la table teachers	40
A.20	Variables qualitatives de la table donors	40
B.1	Test de Spearman project_cost	61

B.2	Tests de Spearman donation_amount	61
B.3	Test du χ^2 / v de Cramér donor_state	61
B.4	Test du χ^2 / v de Cramér poberty_level	62
B.5	Test du χ^2 / v de Cramér status	62
B.6	Régression logistique donation_amount	63
B.7	Régression logistique poverty_level	63
B.8	Régression logistique project_cost	64
B.9	Régression logistique status	64

