Location de vélos partagés

Alexandre Attia - alexandre.attia@wanadoo.fr

31 mai 2016

1 Introduction

Le but de ce test est de construire un modèle pour prédire le nombre de vélos loués par heure dans une ville, en se basant sur des données contextuelles (saison, vacances, météo, heure ...).
Le jeu de données fourni est un ensemble de relevés du nombre de vélos partagés s'étalant entre le 1er janvier 2011 et le 19 décembre 2012, pour les 20 premiers jours de chaque mois, heure par heure.
La première partie du test a pour but de mieux comprendre le jeu de données et d'analyser ces données à l'aide de statistiques descriptives. Il faut pour cela trouver les paramètres influents sur le nombre de vélos.
La deuxième partie est axée sur la construction d'un modèle prédictif et l'analyse de performance de ce modèle.
Dans cet exercice, j'ai joins de nombreux graphiques pour faciliter la compréhension et adopter une démarche assez scientifique que ce soit lors de la description statistique ou de la prédiction. Pour ce test, j'ai utilisé Python 3 avec plusieurs bibliothèques Python : *numpy, pandas, sckit learn* et *matplotlib*.

2 Statistiques Descriptives

Le jeu de données fourni est au format .csv, j'ai donc utilisé la bibliothèque Pandas pour l'utiliser.
Le dataset contient de nombreuses informations : heures, saison, vacances scolaires, la météo, le vent, la température (ressentie et classique), le nombre de locations de vélos (totale, pour les usagers non abonnés et pour les usagers abonnés).
On voit clairement que certains paramètres sont corrélés et donc tous les paramètres ne sont pas forcément utiles. En effet, le nombre total de locations de vélo est la somme des locations pour les abonnés et les non-abonnés. De même, il n'est peut-être pas forcément pertinent d'utiliser la température ressentie et la température normale (0.98 de coefficient de corrélation entre les deux).
La première étape est de vérifier le jeu de données. Le dataset contient 10886 données avec aucune donnée non nulle.

2.1 Influence uniquement du jour et l'heure

Un des facteurs clés que l'on peut observer est le temps, qui se décompose en deux variables pour nous : *hours* et *dayOfWeek* issus de *datetime*.

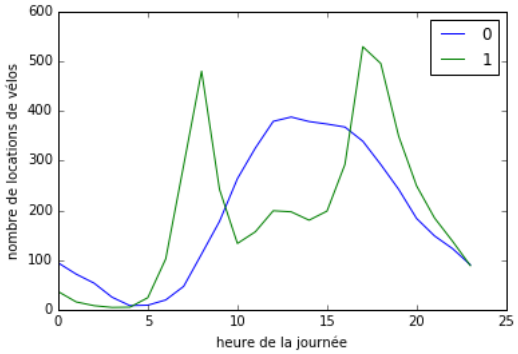

FIGURE 1 – Locations de vélos totales en fonction de l'heure pour chaque jour de la semaine

J'ai d'abord tracé les locations de vélos partagés selon l'heure pour chaque jour. On peut donc constater que l'évolution de location est similaire les jours de semaine alors qu'elle est différente en week-end (*dayOfWeek* =

5 ou 6).

On constate que l'heure est un facteur important pour la location de vélos. En effet, en semaine, il y a un pic d'utilisation le matin vers 8h et le soir vers 18h. Ceci peut logiquement s'expliquer par l'utilisation de ces vélos pour aller travailler donc les pics d'utilisations coïncident avec les horaires de travail. Au contraire, le week-end, l'évolution des locations de vélos a une forme plus gaussienne, diffusée dans l'après midi avec un maximum vers 14h. Cette différence entre week-end et jours de semaine, peut également se visualiser avec la variable *workingday* où l'on peut donc obtenir deux graphiques (un semblable à ceux des jours de semaine et un semblable à ceux du week end).

Pour mieux comprendre cette variation selon l'heure on peut également tracer pour un jour en particulier l'écart-type, sachant que les jours de semaines sont à peu près similaires. Je choisis de faire ce tracé pour le Lundi.

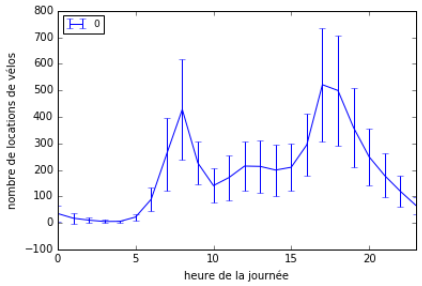

FIGURE 2 – Déviation standard des locations de vélos en fonction de l'heure, pour le Lundi

On peut constater qu'il y a un écart-type assez important au niveau des pics. Il y a donc une volatilité assez importante aux heures de pointes de la demande, explicable par d'autres paramètres à ces heures habituelles de pointes.

2.2 Influence du mois

Nous cherchons à voir l'influence du mois sur la demande en vélos. Plutôt que de s'occuper des saisons, je pense qu'il est plus pertinent de voir la variation de la demande mois par mois.

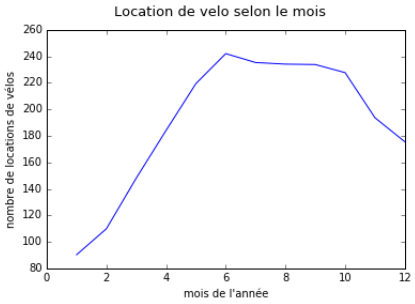

FIGURE 3 – Locations de vélos totale en fonction du mois de l'année

On constate que le mois a une influence assez importante sur la demande. L'été est une période de forte demande pour la location de vélos partagés.

2.3 Influence de la météo

On peut ensuite à l'aide la bibliothèque Pandas calculer la répartition des vélos partagés selon la météo et les 4 valeurs possibles de la météo 1 : Dégagé à nuageux, 2 : Brouillard, 3 : Légère pluie ou neige, 4 : Fortes averses ou neiges. En effet, nous cherchons à prouver que les conditions météorologiques influent sur la location de vélos. Naïvement, nous pouvons penser que de meilleures conditions permettent d'augmenter le nombre de locations.

Au préalable, il faut déjà calculer le nombre d'entrées pour chaque valeur prises par la variable météo. On constate que sur tout le jeu de données, il n'y a qu'une fois de fortes averses. Dans 66 % des cas, la météo est entre dégagé et nuageux, 26 % des données sont lorsqu'il y a du brouillard et les 8 % restants sont pour des

légères averses ou neige.
Nous allons tout de même calculer la moyenne des locations de vélo pour les différentes conditions météorologiques possibles. La valeur pour de fortes averses n'étant pas vraiment une moyenne, elle n'est pas très représentative, j'ai décidé ne pas la mettre sur le graphique suivant.
Ce graphique est tracé sous la forme d'histogramme pour être plus visuel.

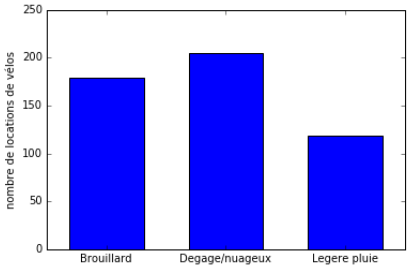

FIGURE 4 – Moyenne des locations de vélos pour différentes conditions météorologiques par rapport à leurs médianes

On constate qu'il y a une dépendance aux conditions météorologiques pour la location de vélos. En effet les locations de vélos diminuent lorsque les conditions se détériorent.

2.4 Influence de la température, du vent, et de l'humidité

Pour chacune de ces trois variables, j'ai calculé la médiane et j'ai calculé la moyenne de la demande en vélos si la valeur de la variable est au dessus ou en dessous de la médiane.
J'ai utilisé la forme d'un histogramme pour faciliter la visualisation et mieux comprendre l'influence de ces trois variables.

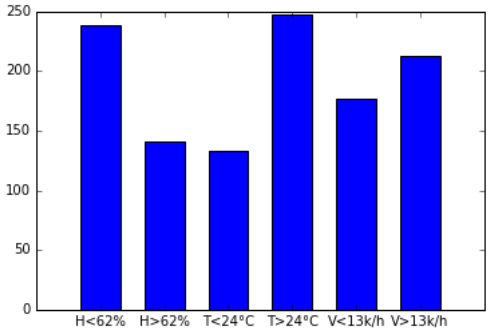

FIGURE 5 – Moyenne des locations de vélos pour différentes conditions météorologiques

On constate donc que le vent n'a pas eu une influence très importante. Même lorsqu'on essaye d'aller à des vitesses de vent plus élevées (35 hm/h), la demande n'évolue pas énormément.
Au contraire, on constate que l'humidité a un impact important sur la demande. Lorsque l'humidité est trop importante, la demande baisse. Ceci est corrélé également à la pluie. Lorsqu'il pleut l'humidité est maximum.
En calculant la matrice de corrélation de ces variables, on constate que les variables *weather* et *humidity* ont un coefficient de corrélation de **0,4**. Ces deux variables sont donc assez corrélées.
La température est le paramètre le plus influent. En effet, lorsque la température est supérieure à 24°C, la demande est bien plus importante que lorsqu'elle est inférieure à 24°C.

2.5 Conclusion et choix des paramètres influents

Nous pouvons en conclure, avant toute modélisation, que les paramètres influents sont **l'heure, le mois, la température** et **la météo**. Certaines variables ont une influence plus légère mais sont tout de même maintenues dans la création du modèle. On constate également que plusieurs variables sont corrélées.

Je n'ai pas utilisé la variable *season*, car celle-ci est corrélée avec d'autres variables. Grâce à la matrice de corrélation, on peut constater que le coefficient de corrélation avec la température est de **0.26** et est de **0,19** avec l'humidité.

De plus, je n'ai pas utilisée la variable saison car il est sûrement plus utile d'utiliser le mois, il peut y avoir des variations dans une même saison. Comme dit précédemment, nous allons utilisée uniquement *temp* et non *atemp*. La variable *holiday* n'est pas très utile quand on regarde l'influence sur *count*, je ne l'ai donc pas utilisée. Les variables *workingday* et *dayOfWeek* sont fortement corrélées. Sachant que *dayOfWeek* est la variable qui contient le plus d'informations, j'ai décidé de ne garder que celles-là.

3 Machine Learning

Nous possédons un assez grand nombre de données labellisées, nous allons donc utiliser des méthodes de machine learning supervisé pour essayer de prédire la demande en vélo partagés.

Nous pouvons modéliser les données sous la forme : $\{(x^{(1)}, y^1),...,(x^{(n)}, y^{(n)})\}$
$x_j^{(i)}$ est les différentes données pour chaque heure j est l'ensemble des paramètres
$y^{(i)}$ est l'output ie la demande en vélo
Nous souhaitons que l'algorithme prédise y.
J'utilise la bibliothèque *scikit learn* pour la construction de différents modèles afin d'en choisir un optimal. La première étape est de diviser aléatoirement notre jeu de données en une partie d'entraînement (67% des données) et une partie test (33% des données). Cette division permettra (excepté pour l'algorithme de Random Forest) a fortiori de mesurer la performance de notre modèle. Nous allons tester plusieurs modèles afin d'en construire un final qui sera le plus performant.

3.1 Construction d'un modèle predictif

3.1.1 Regression Linéaire

Le premier modèle que nous allons construire est une régression linéaire simple. Ce modèle ne devrait pas être très pertinent mais grâce aux coefficients nous pouvons déjà avoir un aperçu des paramètres les plus influents.

Données	Score
Entraînement	0.334
Test	0.343

FIGURE 6 – Score pour la Régression Linéaire simple

Le modèle linéaire étant très peu flexible, le score est en effet assez faible. Mais ce modèle peut également servir de comparaison et permet de comprendre l'importance des différents paramètres. En effet, le coefficient R^2 de détermination n'étant pas si faible, la mesure des valeurs des coefficients de la régression reste pertinent pour comprendre l'influence de chaque paramètre.

3.1.2 Gradient Boosting Regression

Le second modèle est celui du Gradient Boosting, qui est une méthode de boosting où l'on utilise plusieurs modèles predictifs (arbres) créés iterativement qui sont ensuite pondérés pour obtenir la prédiction finale.
Pour visualiser, ce modèle nous allons tracer des learnings curves pour vérifier s'il y a overfitting ou non et pour calculer le score.

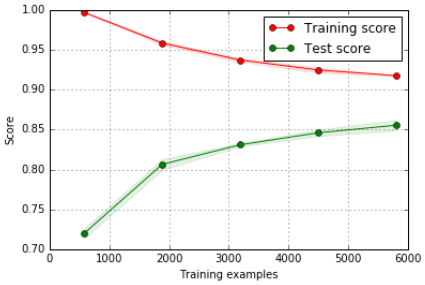

FIGURE 7 – Courbes d'apprentissage pour la régression Gradient Boosting

On constate qu'il y a un écart entre l'erreur sur les données test et sur les données d'entraînement, il y a donc overfitting, la variance de ce modèle doit être trop élevée. Le score est tout de même de : **0.91** pour les données d'entraînement et **0.86** pour les données test. La mesure de la performance, se fait, ici, grâce à au score sur les données test qui ont été choisies aléatoirement à partir des données initiales.

3.1.3 Support Vector Regression

Le troisième modèle est celui des Support Vector Machine pour les régressions, comme l'algorithme pour la classification on ignore les données trop proches de la prédiction. On utilise un kernel linéaire ici.

Données	Score
Entrainement	0.289
Test	0.292

FIGURE 8 – Score pour l'algorithme de SVR

Les resultats obtenus ne sont pas bons mais on peut constater que les données sont semblables.

3.1.4 Random Forest

Le quatrième modèle est celui du Random Forest. C'est un méthode de bagging (on génère des données additionnelles depuis les données d'entraînement en utilisant des combinaisons à répétitions), qui consiste à considérer un ensemble d'arbres appris chacun sur un échantillonage aléatoire de la base d'exemples ; la prédiction se fait par vote majoritaire. L'algorithme effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.
Les valeurs obtenues pour ce modèle sont :

Données	Score
Entrainement	0.972
Test Score	0.844
OOB Score	0.852

FIGURE 9 – Score pour l'algorithme de Random Forest

Là encore, nous pouvons constater un écart entre le score sur les données d'entraînement et les données test, qui est assez important mais la valeur du test score et OOB score reste néanmoins assez élevée. Nous avons changé le paramétrage de ce modèle et utilisé l'estimation Out-Of-Bag. Cette erreur est aussi précise que l'erreur sur un test set de la taille du training set.
Après plusieurs tests, et lectures complémentaires, je me suis arrêté sur un nombre d'arbres de 64 pour avoir une performance optimale sans augmenter le coût de calcul (doubler le nombre d'arbres n'améliore pas sensiblement les performances [2]).

3.2 Performance du modèle

Nous allons tout d'abord comparer les différents modèles présentés :

Modèle	**Training Score**	**Test Score**
Regression Linéaire	0.334	0.343
Gradient Boosting	0.91	0.86
SVR	0.289	0.292
Random Forest	0.972	(OOB) 0.850

FIGURE 10 – Score pour les différents algorithmes

Le modèle choisi est donc l'**algorithme de Random Forest**.
Pour mesurer la performance nous avons utilisé l'erreur Out-Of-Bag. La mesure est assez precise pour mesurer le pouvoir de prédiction de cet algorithme [1]. Le score OOB obtenu est de : **0.85**. Le modèle construit a donc un bon pouvoir prédiction de la demande en vélos partagés. De plus, normalement l'algorithme de Random Forest ne souffre pas d'overfitting si les données ne sont pas bruités, ce qui n'est pas notre cas a priori.

3.3 Améliorations possibles

3.3.1 Apprentissage non supervisé

Pour améliorer ce modèle, nous pourrions par exemple faire une étape de pre-processing avec un machine learning non supervisé. En effet, récemment certains réseaux de neurones profonds, ont des performances améliorées grâce à une étape préliminaire de unsupervised learning. De façon analogique, nous pourrions ici utiliser un algorithme de clustering pour trouver certaines relations entre les données.

3.3.2 Heures de pointe

Lorsque l'on calcule les coefficients de la régréssion linéaire et les paramètres importants de Random Forest ou de Gradient Boosting, constate que l'heure est le paramètre le plus influent sur la demande en vélos. Pour améliorer notre modèle, nous aurions pu, par exemple, ajouter une variable *heure de pointe* qui compterait 4 valeurs possibles :

— 1 si {*hours*= 17-19 et *workingday*=1}
— 2 si {*hours*= 7-9 et *workingday*=1}
— 3 si {*hours*= 12-16 et *workingday*=0}
— 0 sinon

Ces valeurs sont obtenues à partir du graphe de l'influence de l'heure de la partie Statistiques Descriptives.

3.3.3 Localisation des vélos

Nous pourrions également essayer d'obtenir les positions des vélos loués. En effet, la localisation des vélos est sûrement assez importante et doit influer la demande. Nous pourrions ensuite par exemple utiliser un algorithme des K plus proches voisins pour avoir la demande à un endroit donné si on connaît la demande aux alentours. On pourrait également utiliser un algorithme de clustering pour séparer automatiquement la ville selon la demande

Références

[1] Leo Breiman. Out-of-bag estimation. 1996.

[2] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest ? In *Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition*, MLDM'12, pages 154–168, Berlin, Heidelberg, 2012. Springer-Verlag.