**Big Data Analysis: Problem Solving Project**

**Serge GHOMSI : 1654**

**Enseignant: Dr Pierre Yves LABLANCHE**
**January 23, 2016**

Le système "Bike-share" est un système de gestion et de location de vélo présent dans de nombreuse villes occidentales. Ce système a été conçu et utilisé pour la première fois en 1965 dans la ville d'Amsterdam. En 1991 le Danemark a emboité le pas avec sa grande métropole Copenhague, et ensuite ont suivi les autres villes comme Washington. Ce système permet au usagers de louer un vélo pour se rendre à sa destination. Pour une gestion efficace de ce système, aujourd'hui le client doit s'identifier s'il possède déjà un compte ou payer avec une carte de crédit à un point réservé à cet effet. Ce système est devenu très populaire certaines villes dans la mesure où il contribue à la préservation de l'environnement.

Ces structures qui gèrent ces systèmes ont besoin de prédiction sur le nombre d'utilisateurs probables par heure. Cette variable dépend d'un certain nombre de paramètres tels que les heures, les jours, la météo et le climat. Notre travail est donc d'analyser les informations fournies (paramètres et cibles) pour prédire le nombre potentiel de vélo à louer toutes les heures (connaissant uniquement les paramètres). Nous allons utiliser les algorithmes de Machine Learning et plus précisément la Librairie de Scikit Learn. Nous allons nous focaliser sur trois algorithmes de regressions à Savoir, le RandomForestRegressor, Le DecisionTreeRegressor et l'AdaBoosRegressor

## 1. Méthodes et Résultats

Dans ce travail, nous disposons de données issues de la ville de Washington des années 2011et 2012. Les données sont divisés en deux partie, la première qui est celle d'entrainement et la seconde qui est celle de prédiction.

### Data Cleaning

La première étape était celle du Data Cleaining, où nous avons supprimé les années, les secondes et les minutes le dans la base de donnée. Nous avons aussi supprimé les colonnes de demandeur et d'enregistrer. Nous avons séparé nos paramètres de la cible. Nous avons aussi fait un split des données d'entrainement: Sur les 19 jours de données que nous avons tous les mois, nous avons pris les 5 derniers jours (15-19) de chaque mois comme ensemble de validation.

**Visualisation**

Ensuite nous sommes passés à la manipulation des données, la visualisation. Pour une meilleure compréhension et appropriation des données nous avons fait des visualisation en trois dimensions. De celle-ci nous remarquons l'importance du paramètre heure. Une y a une très grande affluence entre 6h et 8h, et entre 17h 19h ce qui correspond bien aux horaires de départ et de retour des usagers de leur différente occupation.

**Selection des features**

Nous avons cherché le niveau d'importance de chaque paramètre afin de leur affecter à chacun un poids pour une meilleure analyse. Nous avons donc utilisé l'algorithme "Extra-TreeRegressor" qui nous a fourni des coefficients d'importance. Nous avons donc ensuite constitué un nouvel ensemble de données ceci en multipliant chaque paramètre par son coefficient d'importance.

**Decision Tree Regressor**

Nous avons donc commencé avec l'algorithme du Decsion Tree Regressor, nous avons tour à tour appliqué avec les paramètres par défaut, puis on a fait une recherche optimale de paramètres et enfin on a fait une comparaison avec l'ensemble de validation et on a trouvé un score de 0.76.

**AdaBoosRegressor**

Ici, nous avons fait le même processus que celui du decision tree et nous avons trouvé un score de validation de de 0.58.

**RandomForestRegressor**

Ici nous avons également suivi le même cheminement et nous avons obtenu un score de 0.78.

**Conclusion**

La méthode que nous avons trouvée plus efficace est celle du RandomForestRegressor, nous avons fait des courbes de d'apprentissage et de validation, qui nous montrent bien une convergence avec le nombre d'éléments que nous avons choisi pour faire GridSearchCV et au niveau du maximum de profondeur. Nous avons donc ensuite après application de l'algorithme ExtraTreeClassifier sur les données de test fait une prédiction à partir du RandomForest. Nous avons enregistré nos résultats sur un fichier CSV qui est joint à ce petit commentaire de travail.