Challenge 2020-2021

Jalal

01/04/2021

Partie prédiction

Etant donné que les données saisies dépendent du temps j'ai opté pour un modèle de prédiction ARIMA (AutoRegressive Integrated Moving Average) applicable aux séries temporelles. J'ai également fais le choix de prendre en compte toutes les données (i.e. depuis Mars 2020). J'aurais bien voulu leur accorder des poids en fonction du contexte lors de leur prélevement (i.e. confinement strict, vacances d'été, couvre-feu...) pour que certaines données soient moins mises en valeur, mais par faute de connaissances et de temps je me suis abstenu.

Pour ce faire, j'ai suivi le cheminement ci-dessous:

1. Importation et nettoyage/réarrangement des données:

J'ai regroupé les lignes par dates (pour avoir une prédiction sur un jour entier) et aggrégé les nombres de vélos par jour. Concernant les colonnes, seules celles contenant la date (*Date*), le total journalier (*Today's total*) et le cumul des comptes (*Grand total*) sont retenues.

2. Etude de la stationnarité:

On remarque que la statistique du test de Dickey-Fuller augmenté (ADF) est assez loin des valeurs critiques et que la p-value est supérieure au seuil (0,05). On en conclut que la série temporelle n'est pas stationnaire. Un moyen de la rendre stationnaire est de différencier la série, ici un décalage d'un pas (jour) est suffisant.

3. Paramètres du modéle ARIMA(p,d,q):

En traçant les graphes des fonctions d'autocorrélaiton et d'autocorrélation partielle, nous avons que: + le coefficient AR, p, est inférieur ou égal à 6 + le coefficient MA, q, esr inférieur ou égal à 2. De plus le coefficient de différentiation d, est égal à 1. A l'aide du package *pmdarima*, et de sa fonction *auto.arima* qui retourne le triplet (p,d,q) pour lequel l'AIC est le minimum, on choisit d'appliquer le modèle ARIMA(5,1,2).

4. Entraînement du modèle sur des valeurs connues (matrices train/test):

En comparant les résultats de notre modèle aux données réelles sur les vingt derniers jours, nous obtenons une erreur quadratique moyenne de 366 et un R^2 de près de 42%. Bien que cela puisse paraître faible, nous nous en contenterons.

5. Prédiction pour la journée du 2 Avril 2021:

Nous estimons le nombre total de vélos passant devant le totem Albert 1^{er} durant la journée du Vendredi 2 Avril à 1490 (estimation effectuée le Jeudi 1 Avril à 19h00).

6. Estimation du nombre de vélos entre 1h00 et 9h00:

Cette dernière partie est celle qui m'a posé le plus de problèmes. C'est notamment du à la façon dont les mesures ont été prises puisqu'elles ne sont ni régulières ni ponctuelles, ce qui les rend difficiles à exploiter afin d'effectuer une prédiction sur un lapse de temps précis. Après de nombreuse tentative je ne suis pas parvenu à écrire un script renvoyant la prédiction du nombre de vélos entre 0h00 et 9h00. Je me suis donc rabattu sur une autre méthode qui est de faire la moyenne des rapports de la première mesure après 9h00 et le total de la journée par jour. J'obtiens ainsi une moyenne de 21,8% et estimons donc le nombre de vélos entre 0h00 et 9h00 à 325 vélos.

Partie visualisation

Pour cette partie du challenge, j'ai voulu créer un gif.

1. Importation et nettoyage/réarrangement des données:

Les données sont importées depuis le site de données opensource de la métropole de Montpellier (https://data.montpellier3m.fr/dataset/comptages-velo-et-pieton-issus-des-eco-compteurs) au format *json* et nous les convertissons en dataframe pour travailler dessus. Nous avons ainsi dix dataframes correspondant chacun à un totem et nous gardons que les variables de temps (*date*), de nombre de vélos (*intensity*) et de localisation (*coordinates*).

2. Création d'une carte intéractive à chaque pas de temps (ici en jours) et enregistrement au format html:

Utilisation du package *folium*.

3. Ouverture du lien, screenshot de l'image et enregistrement au format png:

Utilisation du package *selenium*.

4. Compilation de toutes les images et création du gif:

Cette étape s'effectue directement sur un terminal grace au module *convert* du logiciel *ImageMagick*.

5. Pistes d'amélioration

J'aurais aimé ajouter la date sur chaque image après l'étape 3 pour pouvoir les voir défiler dans le gif.

Liens

Lien du dépôt git: https://github.com/jsakher/Challenge-2020-2021

Lien de la visualisation: https://imagizer.imageshack.com/img923/1988/BP5TEe.gif