Seoul bike Sharing

Python for Data Analysis

Github : https://github.com/Raiyol/python-project

David YANG

Hugo TENG

Les données du Dataset

Donnée	Type
Date	year-month-day
Rented Bike count	Count of bikes rented at each hour
Hour	Hour
Temperature	Celsius
Humidity	%
Windspeed	m/s
Visibility	10m
Dew point temperature	Celsius
Solar radiation	MJ/m2
Rainfall	mm
Snowfall	cm
Seasons	Winter, Spring, Summer, Autumn
Holiday	Holiday/No holiday
Functional Day	NoFunc(Non Functional Hours), Fun(Functional hours)

Objectif

PRÉDIRE LE NOMBRE DE VÉLO À METTRE À DISPOSITION DANS LA VILLE À TOUT INSTANT

ANALYSER LE JEU DE DONNÉE ET Y APPORTER DES MODIFICATIONS SI NÉCESSAIRE

ESSAYER DIFFÉRENT ALGORITHME DE MACHINE LEARNING ET TROUVER LE MEILLEUR MODÈLE

TRANSFORMER LE MODÈLE EN API DJANGO

Analyse des données

Analyse des données

Valeurs nulles

Date	0
Rented Bike Count	0
Hour	0
Temperature(°C)	0
Humidity(%)	0
Wind speed (m/s)	0
Visibility (10m)	0
Dew point temperature(°C)	0
Solar Radiation (MJ/m2)	0
Rainfall(mm)	0
Snowfall (cm)	0
Seasons	0
Holiday	0
Functioning Day	0
Holiday (int)	0
Seasons (int)	0
Functioning Day (int)	0

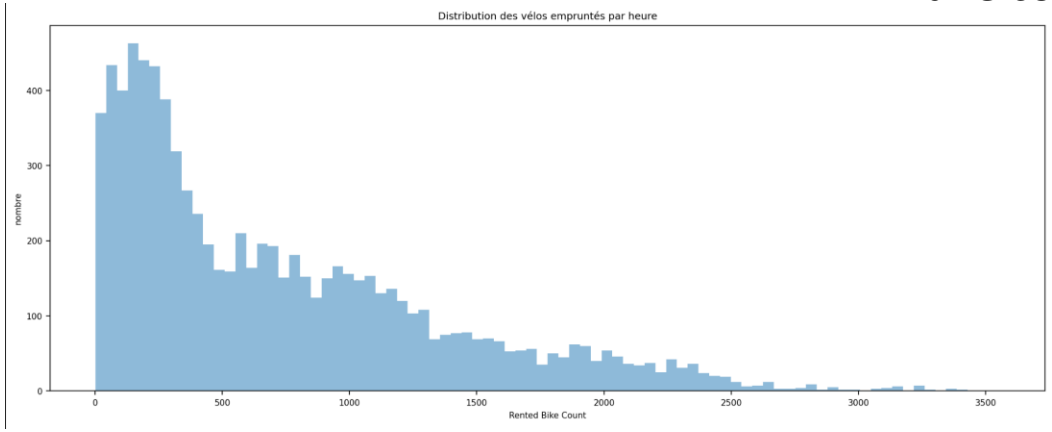
dtype: int64

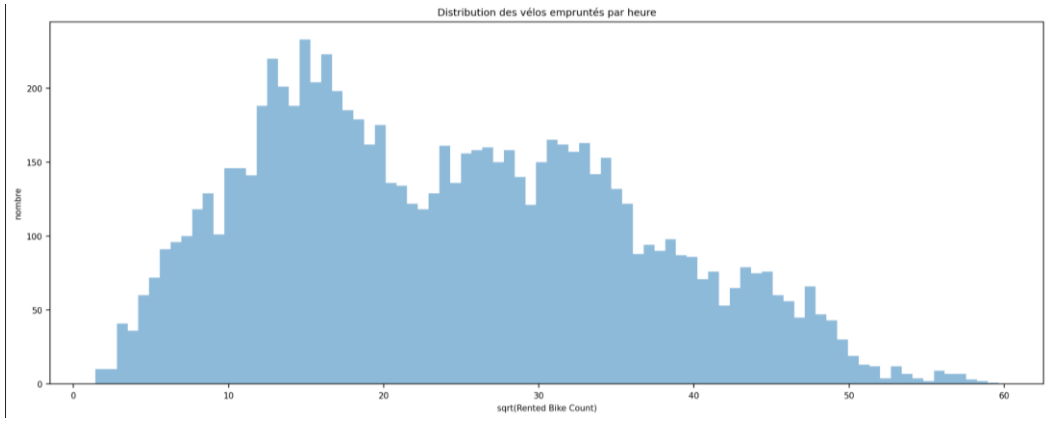
Nous avons aucunes valeurs nulles.
Cela facilite grandement l'analyse des données.

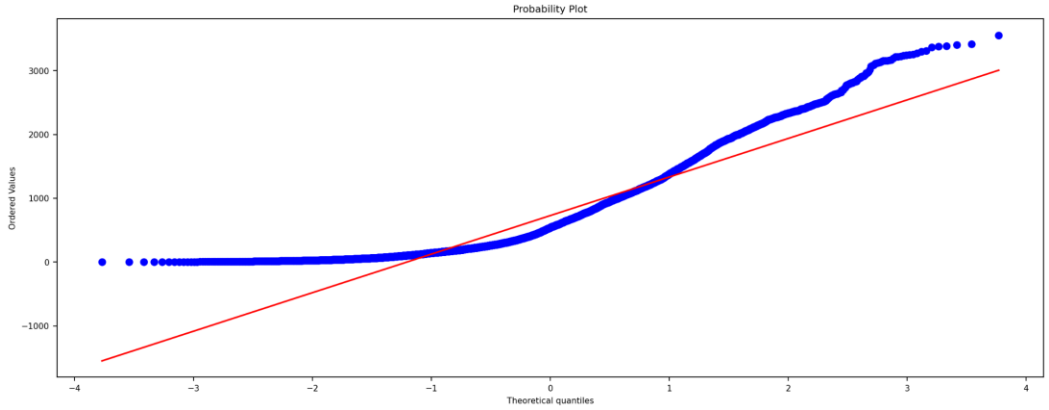
Analyse des données

Nombre de vélo emprunté / objectif

Transformation racine pour obtenir
une courbe pseudo-gaussienne







Analyse des données

Functioning day

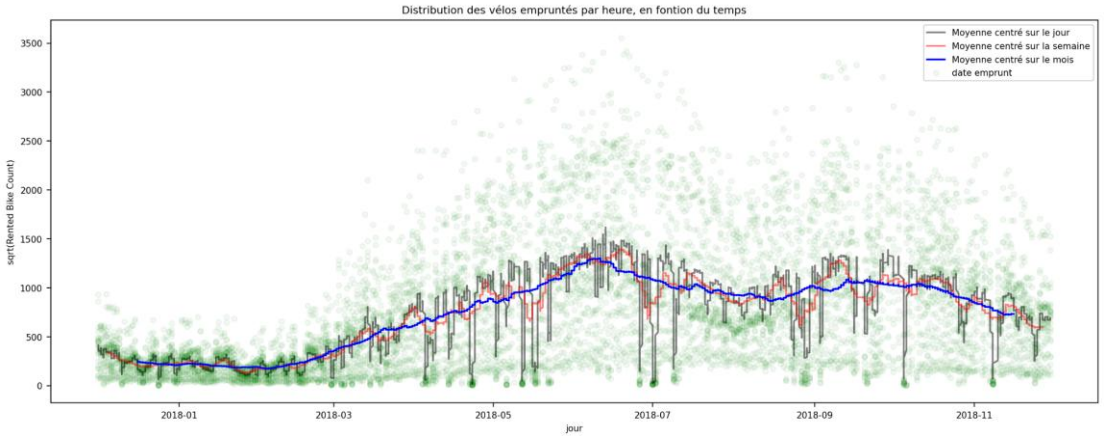
Nous avons commencé par transformer toutes nos variables en variables quantitative afin de pouvoir les analyser et les traiter pour nos modèles. (Seasons en Seasons (int), Date en format date, Functioning day en Functioning day (int), Holiday en Holiday (int)

	Functioning Day (int)	Rented Bike Count sum
0	0	0
1	1	6172314

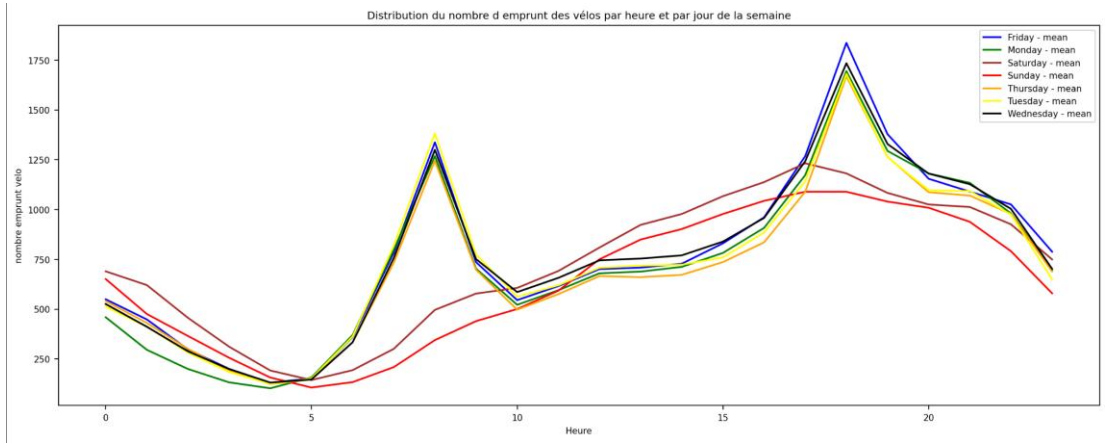
Nous avons remarqué que le nombre de vélos loués est nul lorsque fonctioning day / la station est fermée (quelle surprise !) Nous avons donc décidé de supprimer les lignes où fonctioning day est de 0.

Analyse des données

Nombre de vélo emprunté / objectif



Nous observons une différence en fonction du mois, mais ne pouvons conclure autre chose. Cela demande une analyse supplémentaire faite par la suite.

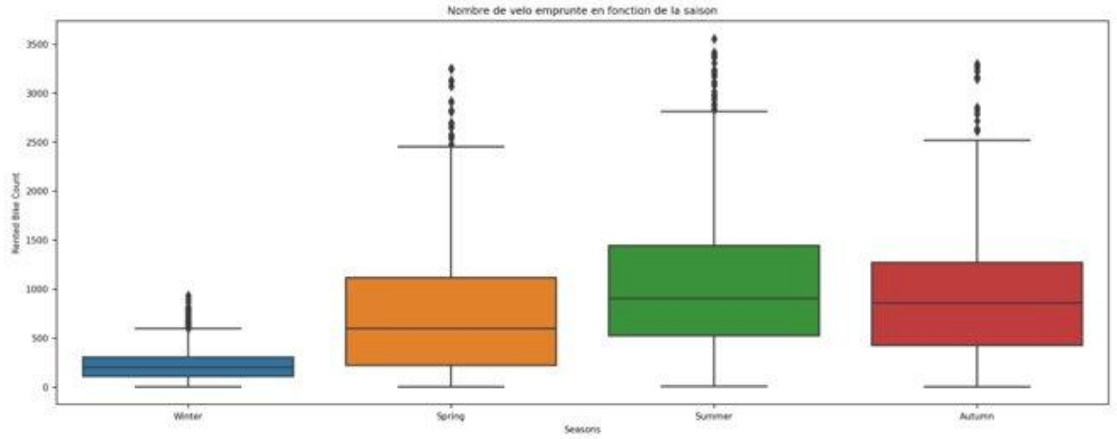


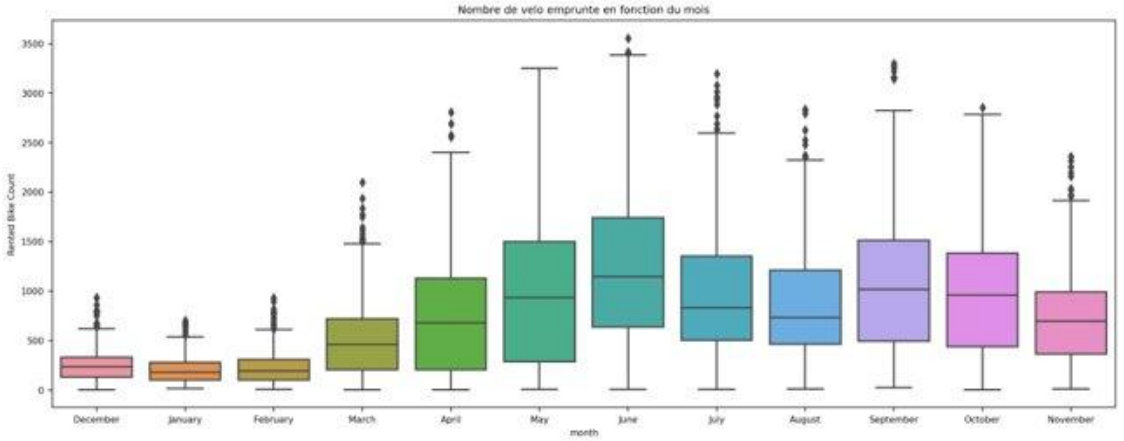
Nous voyons une différence importante si nous sommes en semaine ou le week-end.

Nous ajoutons donc une colonne numérique pour savoir si nous sommes en semaine ou pas, **working_day.**

Analyse des données

Saison et mois





Net déclin en hiver.

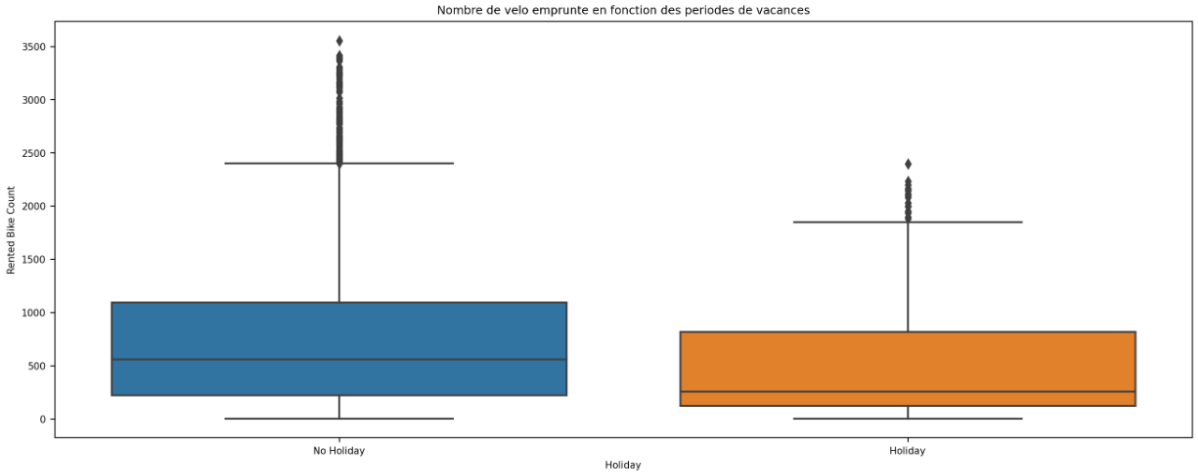
Nous ajoutons donc une colonne numérique pour la saison, **Seasons (int)**.

De même, de Décembre à Février, faible nombre d'emprunt, et inversement le reste du temps.

Nous ajoutons donc pour plus de précision la colonne mois, **month (int)**.

Analyse des données

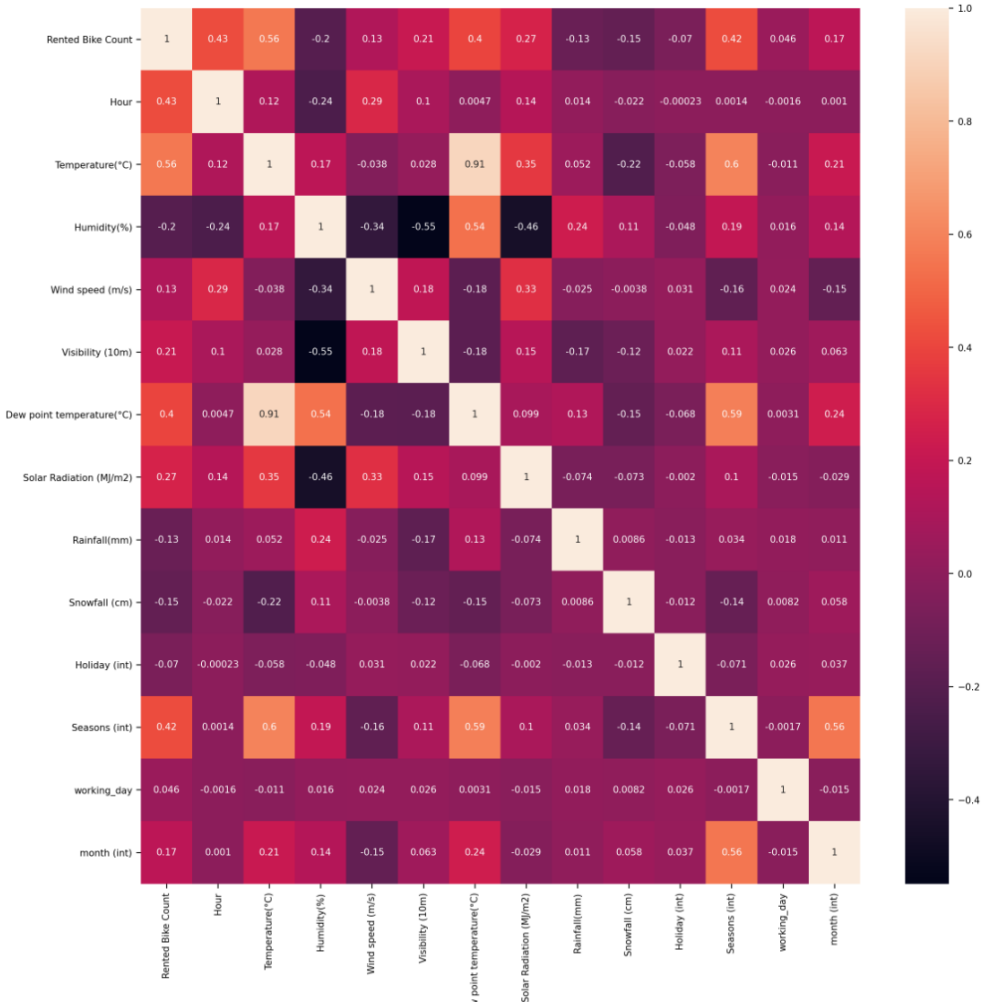
Holiday



Nous observons une faible diminution si nous sommes en période de vacances.
Nous conservons cette donnée.

Analyse des données

Matrice de corrélation



On peut voir que : **Temperature** et **Dew point Temperature** ont une très forte corrélation. Une des deux doit être enlevé, nous avons choisi d'enlever **Dew point Temperature**.

Windspeed, **humidity**, **snowfall** et **Rainfall** ne semble pas avoir un impact fort sur le nombre de vélo.

Visibility et **solar radiation** ont une corrélation forte avec le nombre de vélo

Modélisation

Les algorithmes

Nous avons utilisé 6 algorithmes de modélisation avec des paramétragres différentes :

- Logistic Regression
- Linear Regression
- SVR
- SVC
- K Nearest Neighbour
- Random Forest Classifier

Le jeu de donnée (les entrées)

Nous avons utilisé deux tableaux d'entrées différentes :

- Le premier tableau gardant les données initiales du dataset à l'exception de la date et le functioning day.

	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Holiday (int)	Seasons (int)
0	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	0	1.0
1	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	0	1.0

- Le deuxième tableau provenant de l'analyse des données qui a été faite précédemment.

	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Holiday (int)	Seasons (int)	working_day	month (int)
0	0	-5.2	37	2.2	2000	0.0	0.0	0.0	0	1.0	1	12.0
1	1	-5.5	38	0.8	2000	0.0	0.0	0.0	0	1.0	1	12.0

Le jeu de donnée (les entrées)

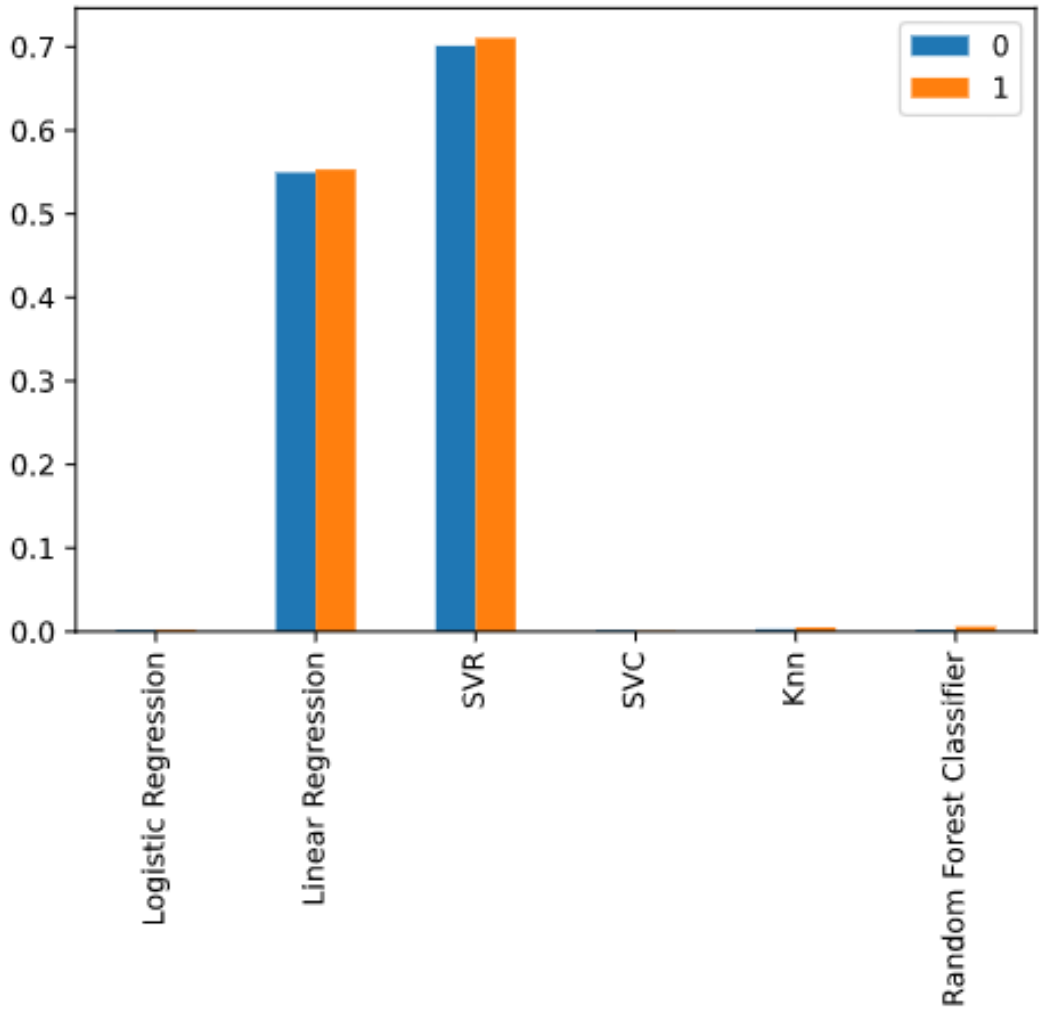
Nous avons ensuite coupé le jeu de donnée en train set et test set avec un découpage de respectivement 2/3 et 1/3.

Les données d'entrées ont ensuite été standardisé.

Dans les deux tableaux, les lignes dont le Functionning day est 0 ont été supprimé. Car si functionning day = 0, le nombre de vélo loué sera aussi forcément 0, cette donnée n'est donc pas intéressante.

Résultats

- En utilisant la fonction score disponible sur chaque modèle dans scikit-learn, ça nous donne une évaluation de la précision des différents modèles, que nous avons ensuite mise sur un graphique pour comparer.
- En bleu, l'entré avec le dataset original et en orange le modifié.
- On remarque que SVR et linear regression sont les algorithmes que marchent le mieux, et que les entrées modifiées performent légerment mieux. Nous allons donc utiliser le modèle SVR avec entrée modifié pour l'API Django



API Django

Création de l'API REST avec Django

Nous avons suivi les étapes dans le cours pour crée le projet Django et l'avons adapté pour notre base de données et modèle.

Pour lancer le serveur, dans un terminal, aller au répertoire "api_django" et utiliser la commande "python manage.py runserver".

Maintenant sur localhost:8000 pour pouvoir utiliser l'API REST.

Les routes

- GET /bikes : Liste de toutes les données
- POST /bikes : Ajouté une ligne de donnée
- GET /bike/x/ : x un numéro, pour une ligne de donnée précise
- DELETE /bike/x/ : x un numéro, supprimé la ligne avec ce numéro
- POST /predict/ : Prédire le nombre de vélo avec le modèle que nous avions fait.

Corps des requêtes post

Les deux routes avec une requête POST doivent avoir un body de ce format :


```
{
    "Hour": 2,
    "Temperature" : 24,
    "Humidity": 40,
    "WS" : 1,
    "Visibility" : 1500,
    "SR" : 0,
    "Rainfall" : 0,
    "Snowfall" : 0,
    "Holiday" : 0,
    "Seasons" : 3,
    "WD" : 1,
    "Month" : 9,
    "RBC" : -1
}
```


Vous pouvez changer les données en fonctions de ce que vous voulez, ne pas toucher à RBC si c'est pour la requête de prédiction.