Python for Data Analysis

Seoul Bike Prediction

Theo Zangato
Youssef Zemali

IBO 5

Contexte/ Problématique

- ***Introduction des vélos de location dans de nombreuses villes urbaines pour améliorer le confort de la mobilité***

- ***Rendre le vélo de location disponible et accessible au public au bon moment***
- ***Réduire le temps d'attente***
- ***Offre stable de vélos de location***

- ***Données : informations météorologiques, date, nombre de vélos loués par heure***

OBJECTIFS

- Prédire le nombre de vélos loués à Séoul pour un jour et une heure donnée

- Problème de régression

Différentes étapes :

1. Présentation des données
2. Analyse des données
3. Features Engineering
4. Modèle de prédiction
5. Résultats
6. Déploiement du modèle

1. PRÉSENTATION DES DONÉES

01
FEATURES

Les données d'entrée de notre modèle qui nous permettent de prédire la Target

Date	object
Rented Bike Count	int64
Hour	int64
Temperature(°C)	float64
Humidity(%)	int64
Wind speed (m/s)	float64
Visibility (10m)	int64
Dew point temperature(°C)	float64
Solar Radiation (MJ/m2)	float64
Rainfall(mm)	float64
Snowfall (cm)	float64
Seasons	object
Holiday	object
Functioning Day	object

02
TARGET

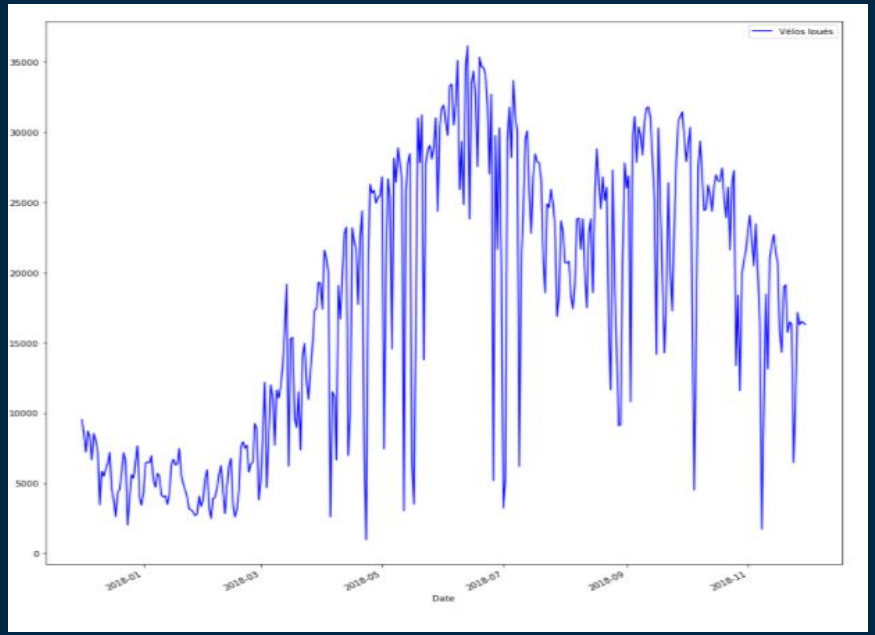
La donnée que l'on souhaite prédire : **le nombre de location de vélos** ici qui est une variable quantitative

Dimensions: 8760 lignes , 14 colonnes

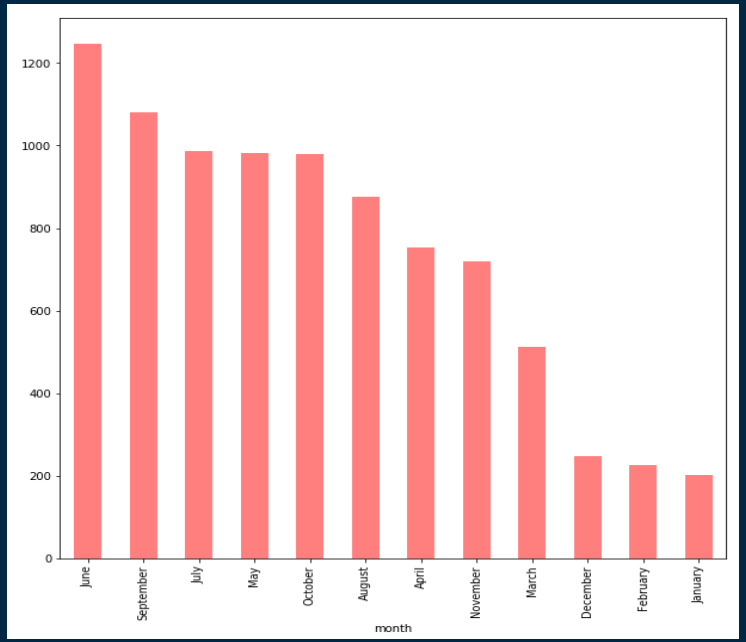
Features: 12 variables quantitatives / 1 variable qualitatif

2. ANALYSE DES DONNÉES

Location en fonction de la date:



Par mois :



Graphiques complémentaires: les locations sont élevés durant les mois de Mai-Juin-Juillet-Septembre et chutent petit à petit en s'éloignant de ces mois.

2. ANALYSE DES DONNÉES

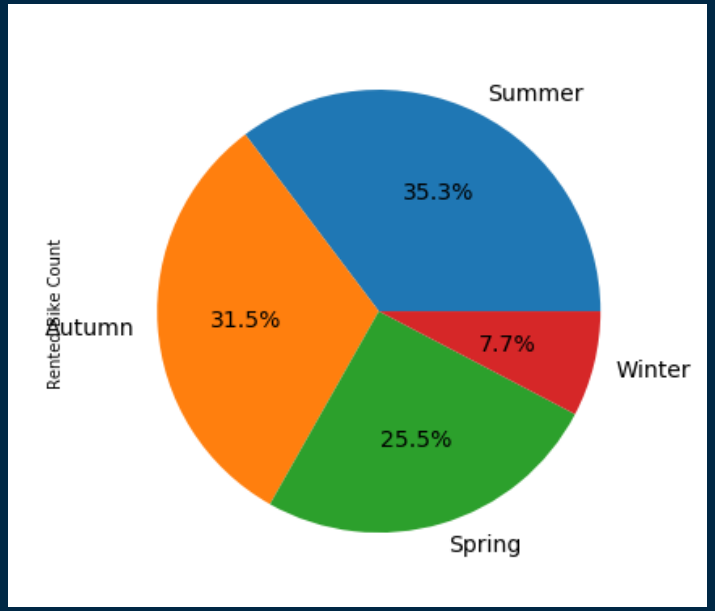
Location de vélo moyenne selon l'heure et le jour de la semaine :

week_day	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
Hour							
0	549.882	459.577	690.1	651.373	540.86	512.917	526.06
1	448.608	296.173	621.06	475.961	431.38	412.083	412.02
2	298.392	199.865	456.56	365.49	301.04	282.042	288.52
3	200.373	132.769	311.78	256.235	191.68	184.938	199.5
4	130.431	103.154	192.22	157.078	123.7	124.688	131.76
5	153.373	157.423	144.4	106.863	148.02	152.479	147.34
6	360.02	370.346	194.04	134	333.96	362.688	333.98
7	783.569	797.481	300.922	209.255	733.2	817.792	756.56
8	1337.96	1268.9	496.667	344.922	1244.4	1382.67	1300.06
9	736.118	703.808	578.529	440.098	698.22	775.104	751.56
10	545.706	522.231	606.882	500.941	497.66	562.188	586.02
11	616.02	595.212	691.843	592.961	575.52	620.312	657.34
12	701.451	680.212	809.627	750.627	666.4	708.042	745.44
13	708.588	688.885	923.392	849.608	660.8	718.75	754.24
14	728.51	712.538	978.02	902.392	672.48	723.479	770.26
15	830.608	782.635	1067.27	977.922	736.62	760.125	839.46
16	961.255	908.269	1137.86	1043.98	835.6	884.646	958.12
17	1268.24	1173.17	1232.41	1089.61	1089.52	1142.4	1242.74
18	1836.76	1695	1181.73	1089.47	1669.18	1681.71	1734.82
19	1378.47	1294.48	1083.69	1040.22	1264.5	1262.67	1329.24
20	1154.76	1181.25	1025.71	1009.25	1087.18	1097.58	1180.56
21	1090.24	1134.25	1013.1	938.706	1070.28	1092.98	1127.54
22	1025.98	980.904	926.176	791.176	978.2	975.25	1003.64
23	788.686	694.981	749.784	579.784	689.58	649.896	702.34

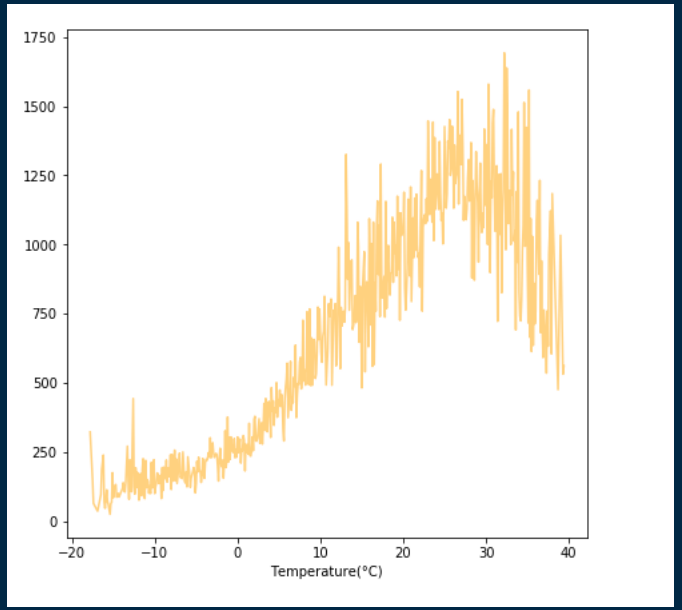
- *On peut voir des pics réguliers à 8h et 18h tout au long de la semaine sauf le samedi et dimanche*
- *Le week-end , les locations sont plus élevés l'après-midi, entre 12 et 18h*
- *Mis en évidence des habitudes quotidiennes des Séoulites concernant la location de vélo*

2. ANALYSE DES DONNÉES

Pourcentage de location selon la saison:



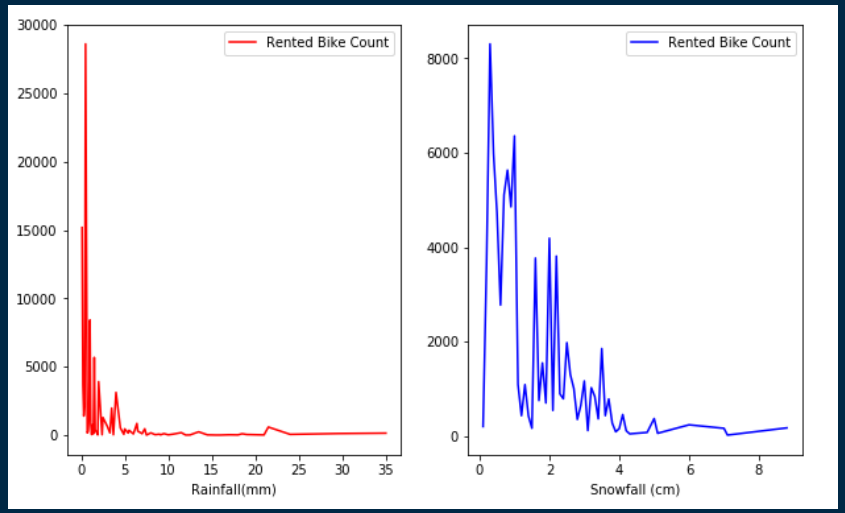
Selon la température :

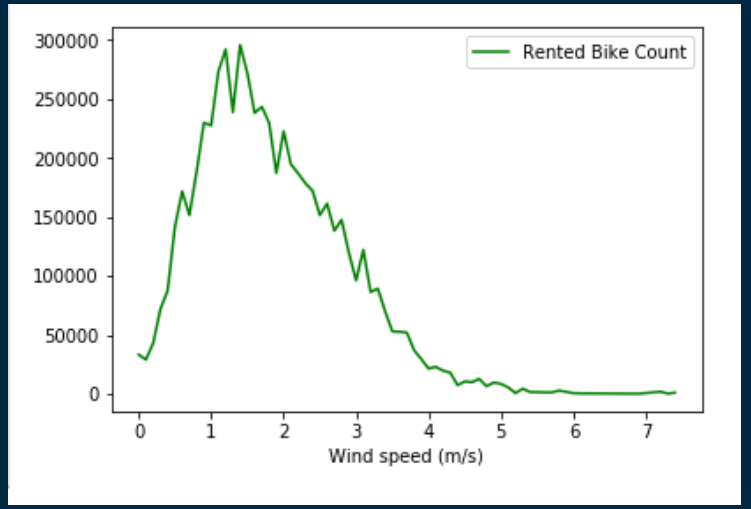


Les locations augmentent quand les températures sont douces, c'est-à-dire entre 20 et 30° C. Ceci est confirmé par le camembert sur le pourcentage des locations selon la saison: l'hiver montre une baisse significatif du pourcentage de location de vélo.

2. ANALYSE DES DONNÉES

Location de vélo en fonction de différentes conditions climatique:





On voit que pour ces 3 premiers graphiques, la location de vélos décroît fortement en cas de fortes pluies, chutes de neiges ou de vents importants. La météo sera sans doute un feature important pour notre modèle. Afin de voir d'autres analyses graphiques, merci de consulter le fichier Project.ipynb.

3. FEATURES ENGINEERING

- ✓ *Suppression des lignes dont le feature « Functionning Day » est égal à « No » : ce feature signifie que la location de vélo ne fonctionnait pas un jour donné donc pas de location par définition*
- ✓ *Suppression de la colonne « Functionning Day » dans le même temps*

- ✓ *Feature Date : transformation du feature Date en 4 features qui sont Jour, Mois, Année , Jour de la semaine*
- ✓ *Suppression de la colonne « Date» dans le même temps*

- ✓ *One Hot Encoding du feature « Season » qui devient donc 4 features différent : Autumn, Winter, Spring et Summer*
- ✓ *Suppression de la colonne « Season» dans le même temps*

3. FEATURES ENGINEERING

Création de 2 datasets avant la prédiction du modèle :

- *1er dataset : features « Mois » et « Jour de la semaine » Label Encoder*
- *2ème dataset : features « Mois » et « Jour de la semaine » One Hot Encoder*

Normalisation des features suivant :
'Temperature', 'Humidity', 'Windspeed', 'Visibility', 'DewPointTemperature', 'SolarRadiation', 'Rainfall', 'Snowfall'

MinMaxScaler :

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

4. MODÈLE DE PRÉDICTION

Modèle de régression *:*

- *LinearRegression*
- *RandomForestRegressor (Fine tuned with GridSearch)*
- *GradientBoostingRegressor (Fine tuned with GridSearch)*
- *XGBRegressor (Fine tuned with GridSearch)*

Fonctions d'évaluations :

$$MSE = \frac{\sum_{i=1}^{n}(f(x_i) - y_i)^2}{n}$$

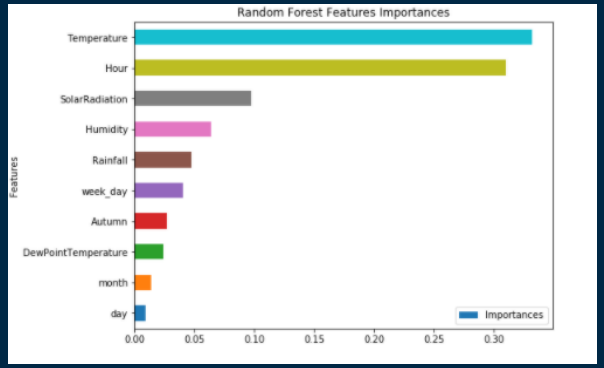
$$RMSE = \sqrt{MSE}$$

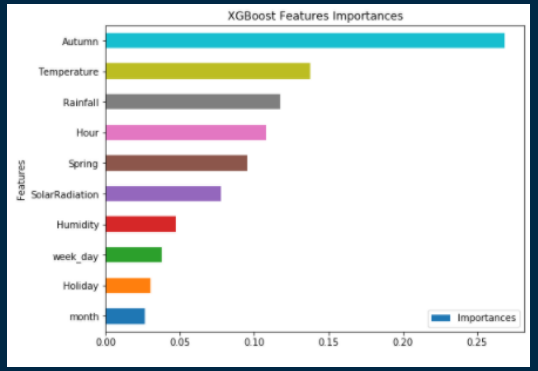
$$R^2 = 1 - RSE$$

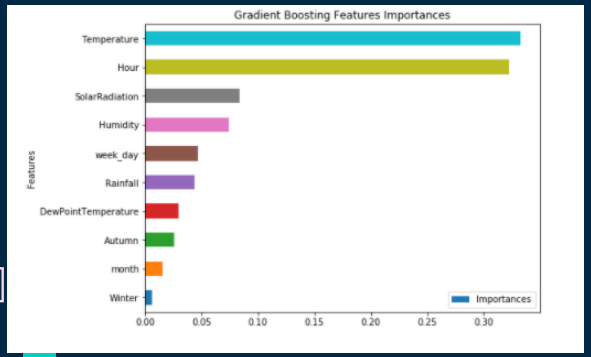
$$RSE = \frac{\sum_{i=1}^{n}(f(x_i) - y_i)^2}{\sum_{i=1}^{n}(\bar{y_i} - y_i)^2}$$

5. RÉSULTATS

Features d'importances :



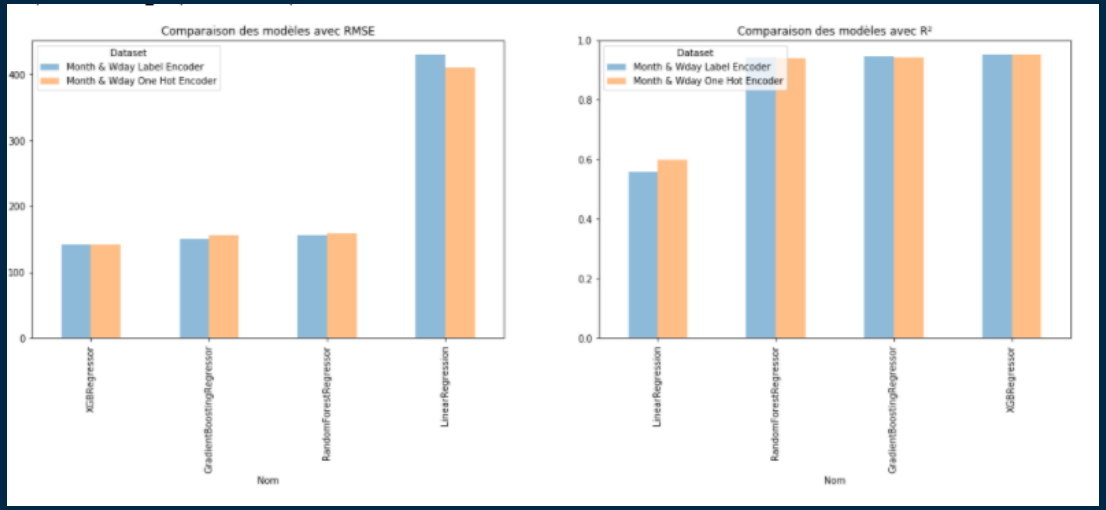




Les 3 modèles ont dans l'ensemble les mêmes features d'importance notamment la température, l'heure et la pluie.

5. RESULTATS

	Nom	Dataset	MSE	RMSE	R²	Set Hypermaters
0	LinearRegression	Month & Wday Label Encoder	185337.818964	430.508791	0.559110	None
1	LinearRegression	Month & Wday One Hot Encoder	168260.795038	410.196045	0.599733	None
2	RandomForestRegressor	Month & Wday Label Encoder	24530.028152	156.620650	0.941647	{'max_depth': None, 'n_estimators': 200}
3	RandomForestRegressor	Month & Wday One Hot Encoder	25076.564822	158.355817	0.940347	{'max_depth': None, 'n_estimators': 200}
4	GradientBoostingRegressor	Month & Wday Label Encoder	22466.747915	149.889119	0.946555	{'learning_rate': 0.1, 'max_depth': 5, 'n_esti...
5	GradientBoostingRegressor	Month & Wday One Hot Encoder	24139.639032	155.369363	0.942575	{'learning_rate': 0.1, 'max_depth': 5, 'n_esti...
6	XGBRegressor	Month & Wday Label Encoder	20047.733935	141.590021	0.952310	{'learning_rate': 0.1, 'max_depth': None, 'n_e...
7	XGBRegressor	Month & Wday One Hot Encoder	20257.230407	142.327898	0.951811	{'learning_rate': 0.1, 'max_depth': None, 'n_e...



Settings des modèles et comparaison des modèles :

On peut voir que les algorithmes utilisant des arbres de décisions (RandomForestRegressor, GradientBoostingRegressor, XGBRegressor) sont très performant dans notre cas.
Meilleur modèle : XGBoost Regressor

R^2=0,952310 (variance très bien expliqué)
RMSE=141,59

Ceux sont des résultats plutôt satisfaisant !

6. DÉPLOIMENT DU MODÈLE

- *Enregistrement du modèle, de l'encodeur et du normaliser sous forme de fichier Pickle*

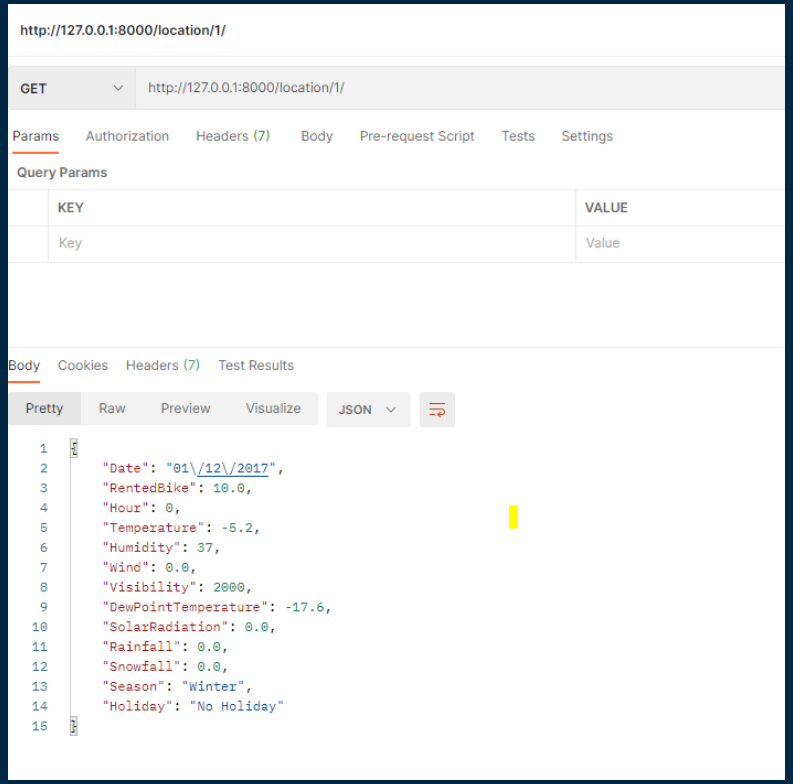
- *Implémentation d'une API Rest Django*
- *Création du modèle*

- *Pré-Processing pour les nouvelles entrées pour rester en adéquation avec le modèle*

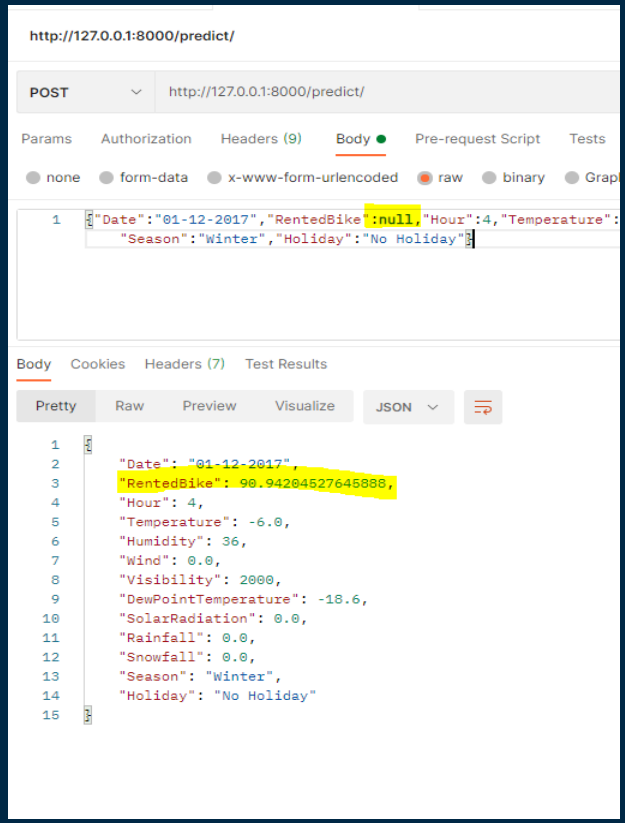
- *Utilisation de Postman Agent pour interroger notre API et vérifier son bon fonctionnement*

6. DÉPLOIMENT DU MODÈLE

GET :



POST :



Thank You

To see more about this project,
please look at this file Project.ipynb