Seoul Bike Sharing Demand

Clément FLORVAL - Anis OUFKIR - Elias FERREIRA

Sommaire

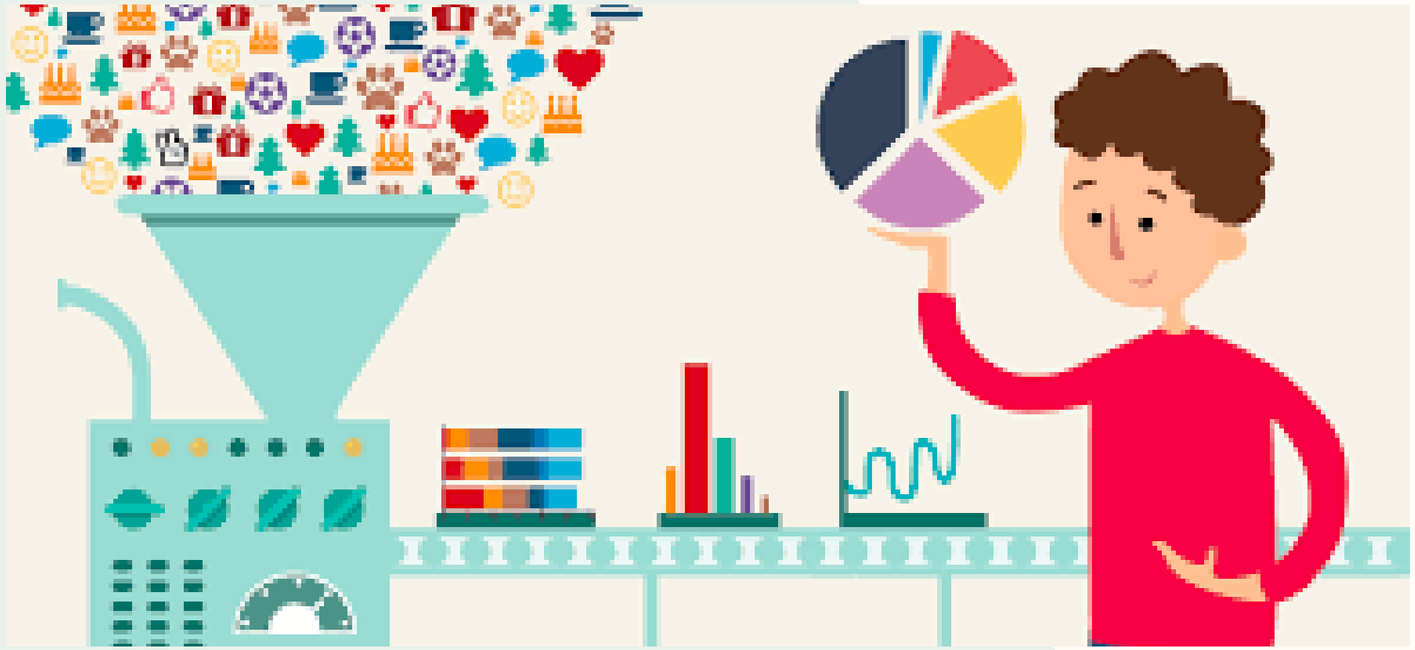
Problème

Optimiser un système de location de vélos à Séoul

Comment prédire efficacement la demande fluctuante de vélos à chaque heure de la journée

Problème intéressant pour nous car il nous met en situation d'un problème qu'on pourra rencontrer en entreprise plus tard.

Prétraitement des données



0	Date	8760 non-null	object
1	Rented Bike Count	8760 non-null	int64
2	Hour	8760 non-null	int64
3	Temperature(°C)	8760 non-null	float64
4	Humidity(%)	8760 non-null	int64
5	Wind speed (m/s)	8760 non-null	float64
6	Visibility (10m)	8760 non-null	int64
7	Dew point temperature(°C)	8760 non-null	float64
8	Solar Radiation (MJ/m2)	8760 non-null	float64
9	Rainfall(mm)	8760 non-null	float64
10	Snowfall (cm)	8760 non-null	float64
11	Seasons	8760 non-null	object
12	Holiday	8760 non-null	object
13	Functioning Day	8760 non-null	object

La donnée à notre disposition

8760 lignes qui représente les features à chaque heure entre le 1er décembre 2017 et le 30 novembre 2018

- **4 variables catégoriques**
- **10 variables discrètes**

Variable cible = Rented Bike Count

Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)
01/12/2017	254	0	-5.2	37	2.2	2000	-17.6
01/12/2017	204	1	-5.5	38	0.8	2000	-17.6
01/12/2017	173	2	-6.0	39	1.0	2000	-17.7
01/12/2017	107	3	-6.2	40	0.9	2000	-17.6
01/12/2017	78	4	-6.0	36	2.3	2000	-18.6

Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0.0	0.0	0.0	Winter	No Holiday	Yes
0.0	0.0	0.0	Winter	No Holiday	Yes
0.0	0.0	0.0	Winter	No Holiday	Yes
0.0	0.0	0.0	Winter	No Holiday	Yes
0.0	0.0	0.0	Winter	No Holiday	Yes

Données brutes sans transformation

Pas de valeur manquante !

Number of null values per column :

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

Traitement et transformation de la donnée :

Date	Rented Bike Count	Hour	Temperature	Humidity	Wind speed	Visibility	Dew point Temperature
2017-12-01	254	0	-5.2	37	2.2	2000	-17.6
2017-12-01	204	1	-5.5	38	0.8	2000	-17.6
2017-12-01	173	2	-6.0	39	1.0	2000	-17.7
2017-12-01	107	3	-6.2	40	0.9	2000	-17.6
2017-12-01	78	4	-6.0	36	2.3	2000	-18.6

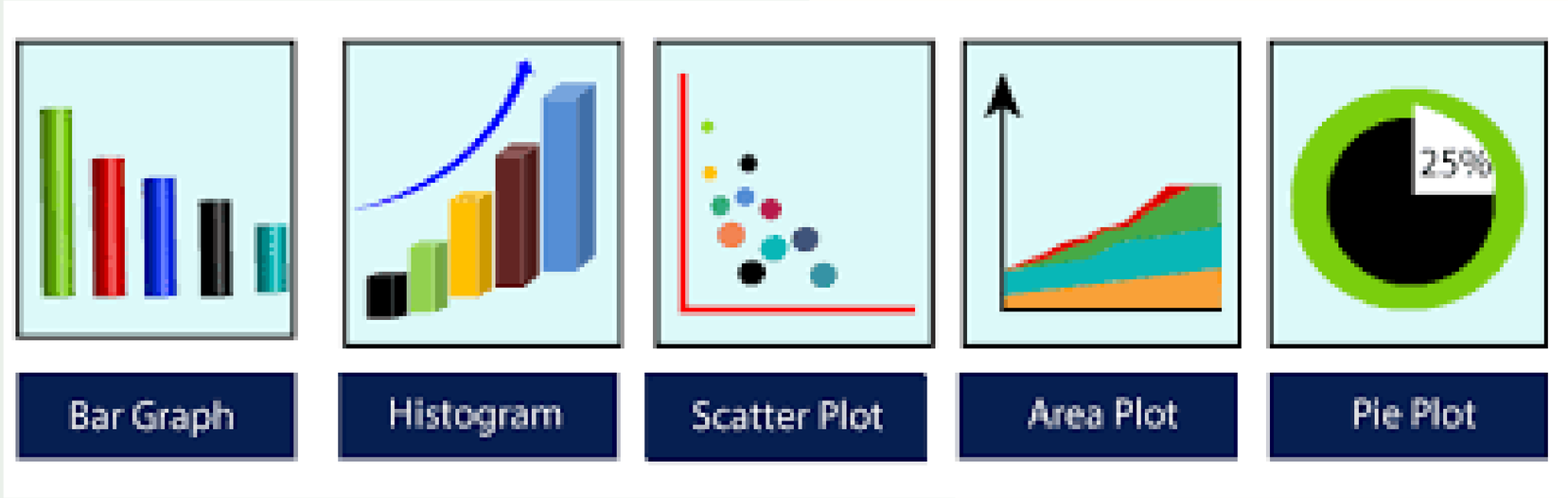
Datetime

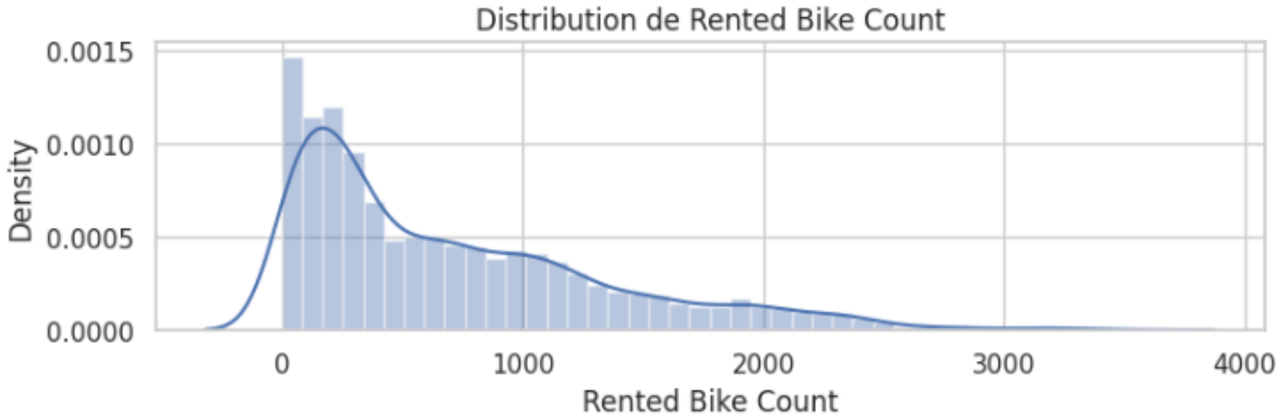
Solar Radiation	Rainfall	Snowfall	Seasons	Holiday	Functioning Day	Weekend
0.0	0.0	0.0	4	0	1	0
0.0	0.0	0.0	4	0	1	0
0.0	0.0	0.0	4	0	1	0
0.0	0.0	0.0	4	0	1	0
0.0	0.0	0.0	4	0	1	0

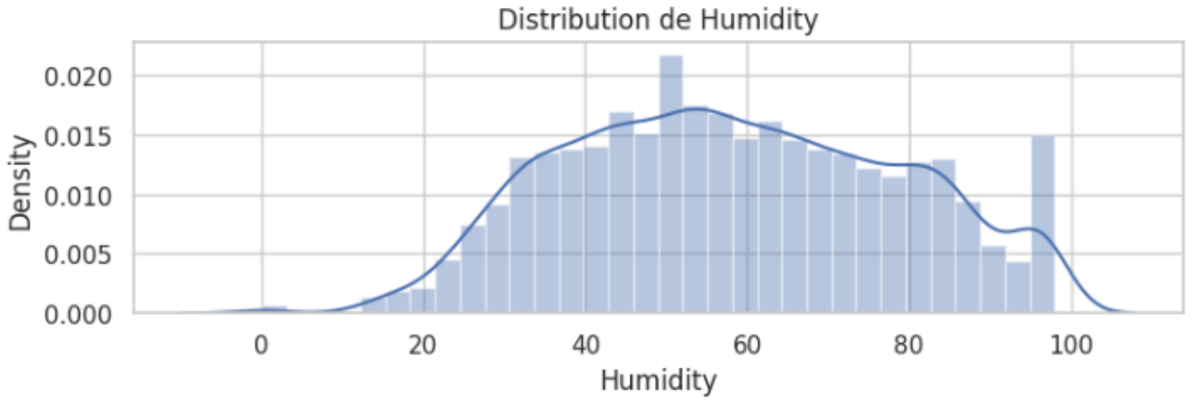
Nouvelle colonne ‘Weekend’

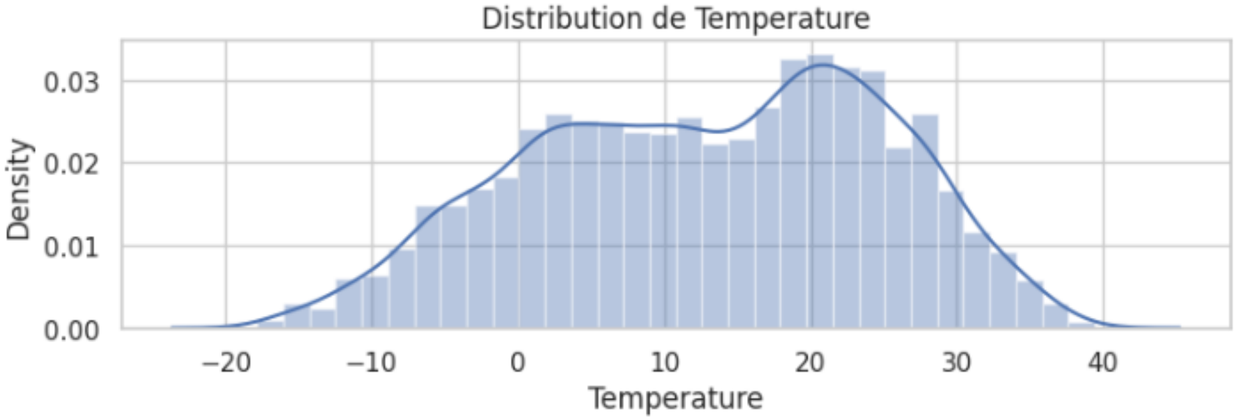
Encodage

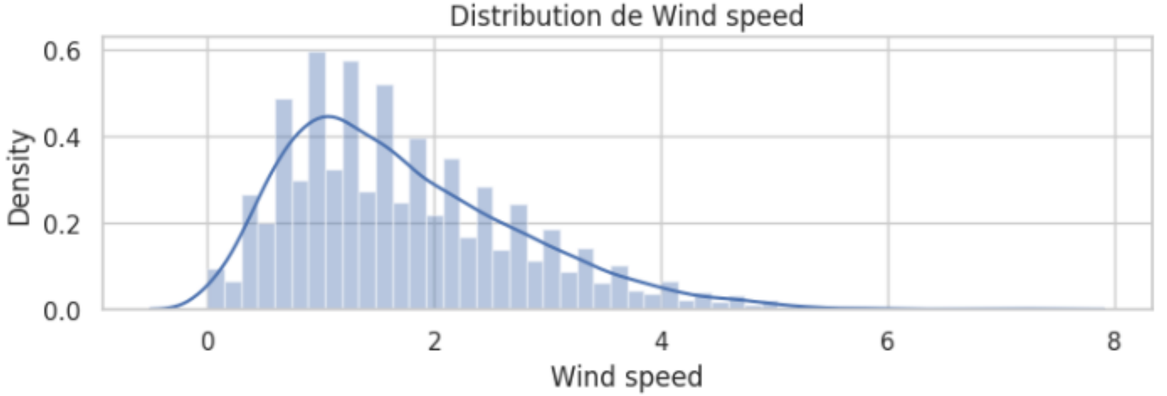
Quelques visualisations pour comprendre comment les données se comporte



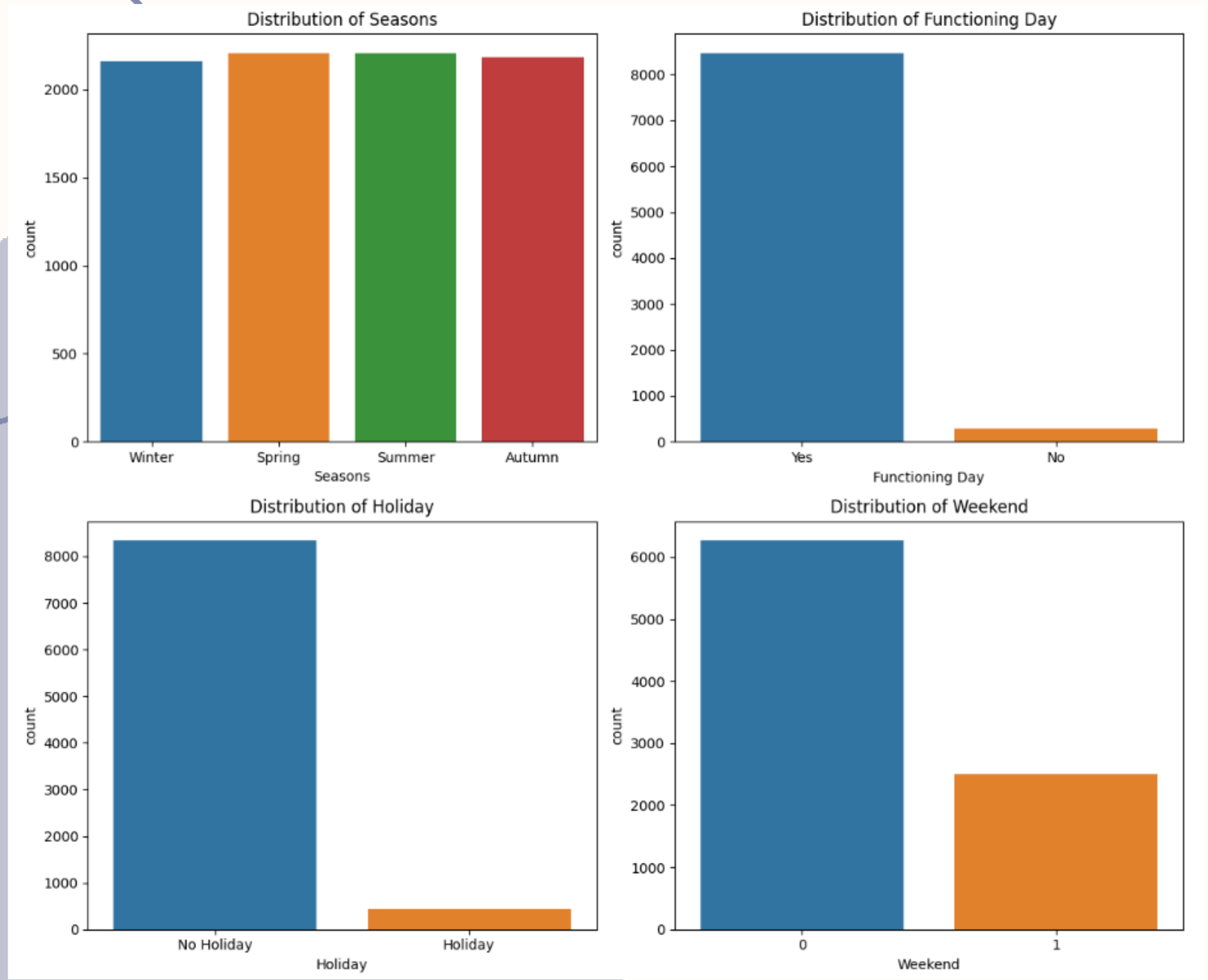




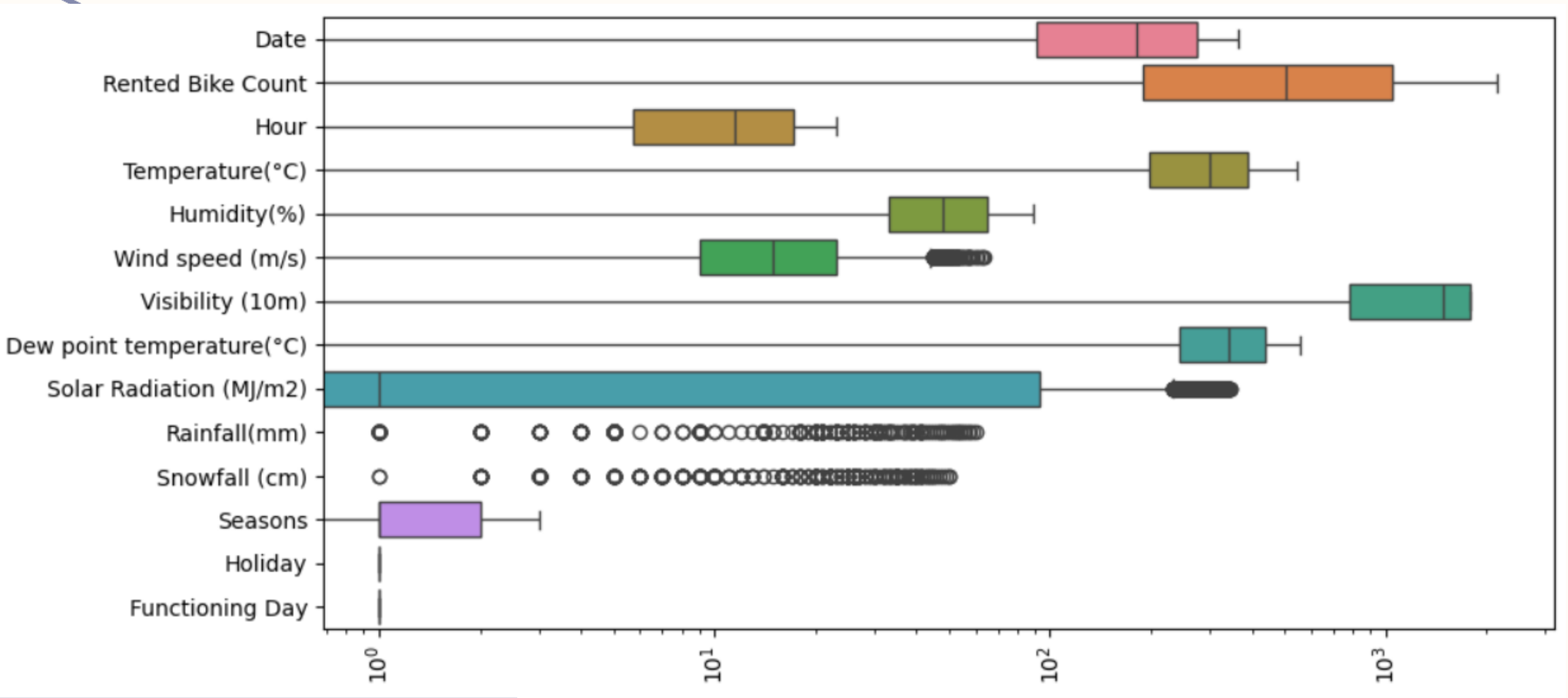




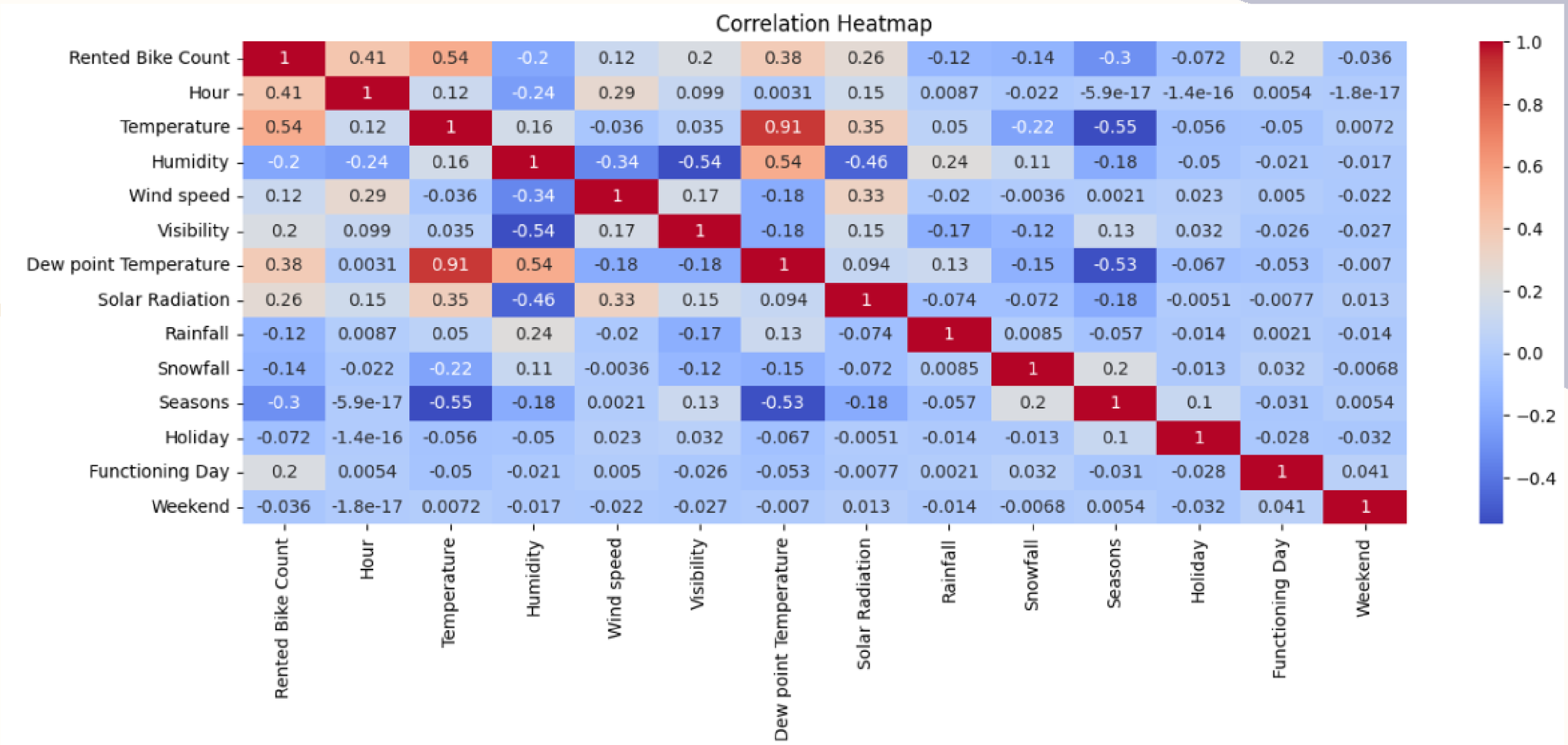
Observons dans un premier temps la distribution de ces variables numériques



Distribution des valeurs prises par chaque variable catégorique



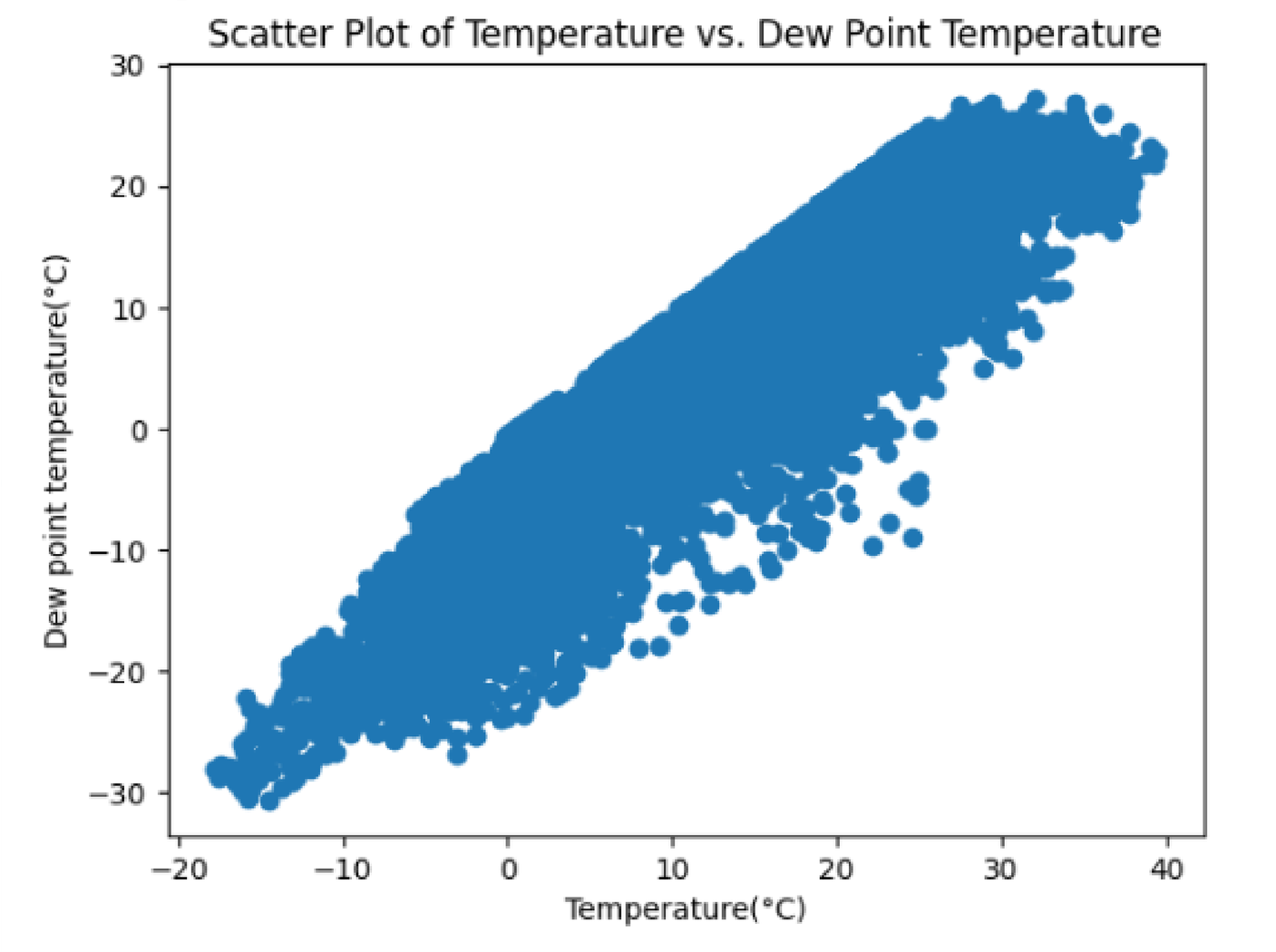
Variance des caractéristiques



Les corrélations

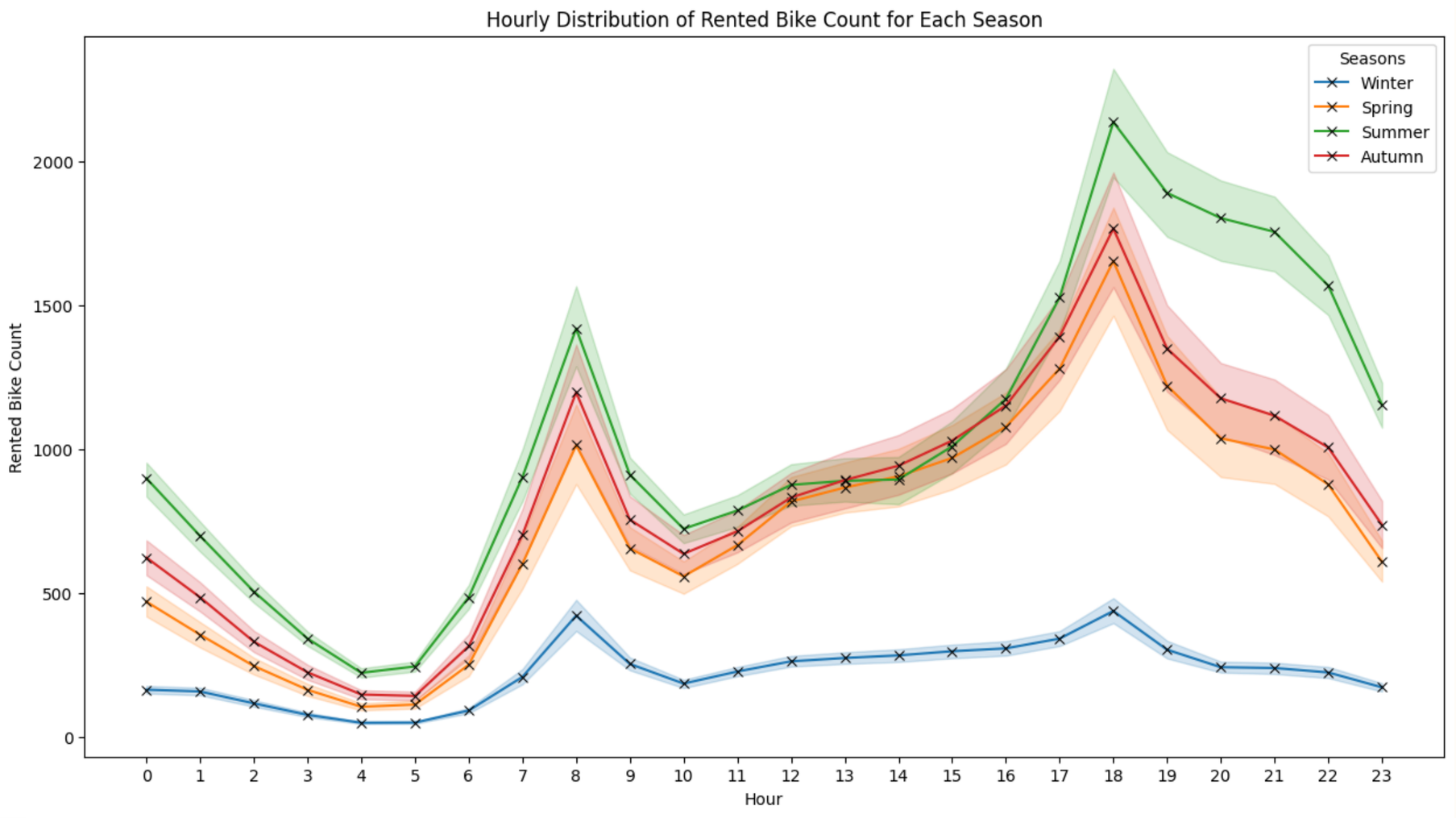
Les features les plus corrélés avec la variable cible sont la température et l'heure

Très forte corrélation entre la température et la température de rosée



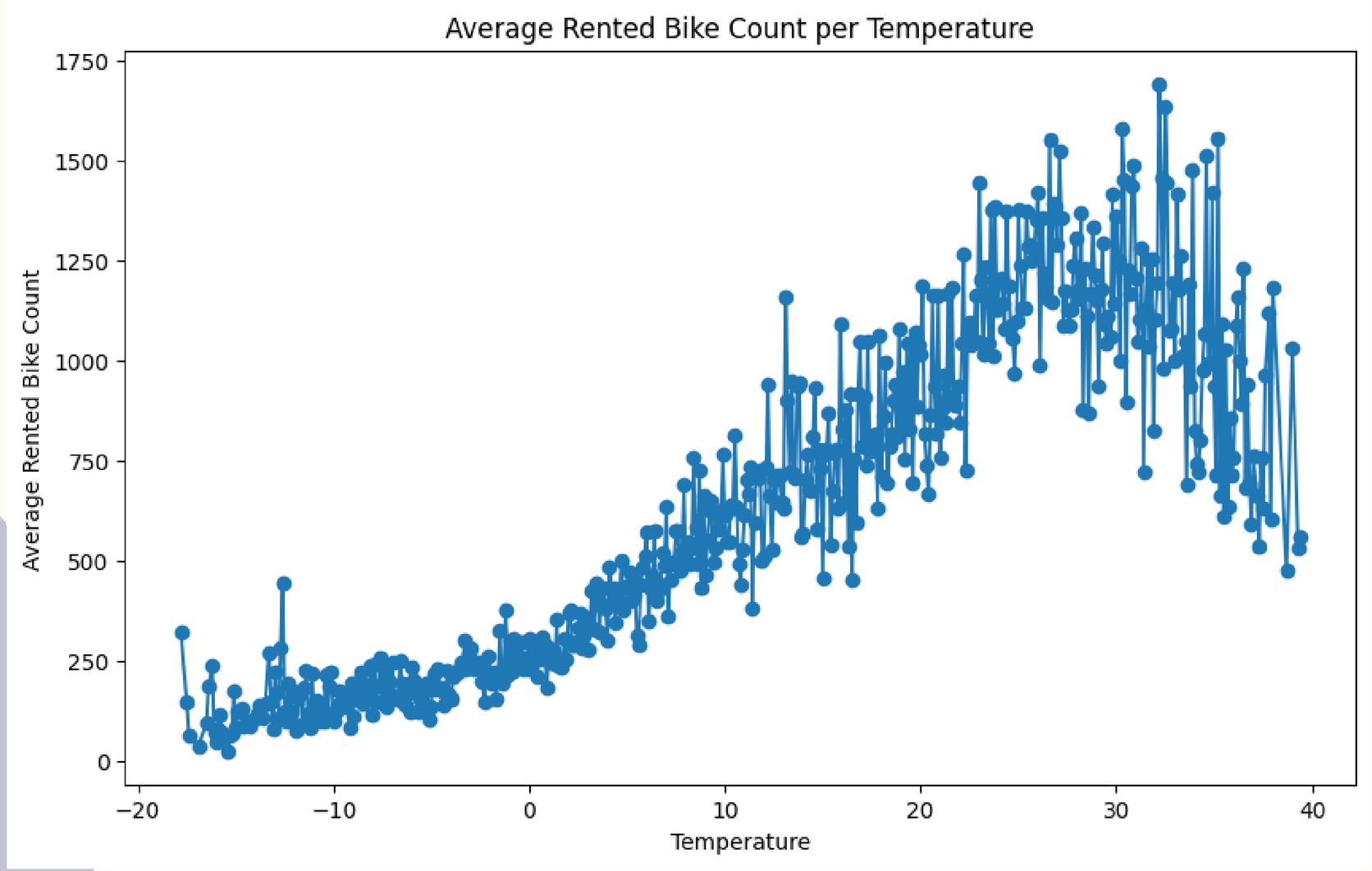
Observons de plus plus près cette corrélation

Suppression de la température de rosée dans la dataframe



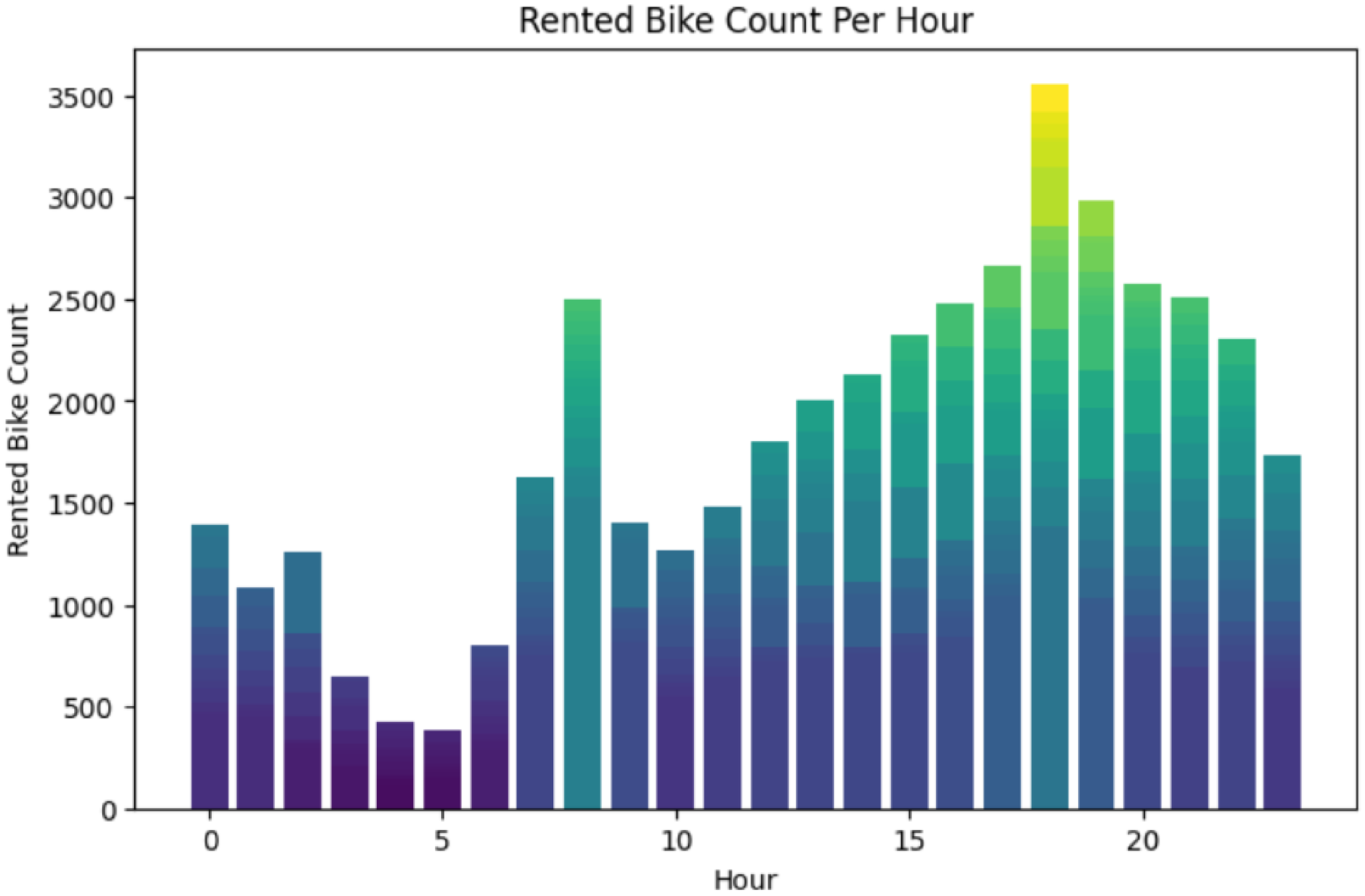
Les conditions météorologiques semblent influencer les locations de vélos

La demande de vélos diminue en hiver, puis augmente au printemps et en automne. Elle atteint son pic en été, probablement en raison de meilleures conditions météorologiques et d'une augmentation des activités extérieures.



La température est déterminante pour la demande de vélos

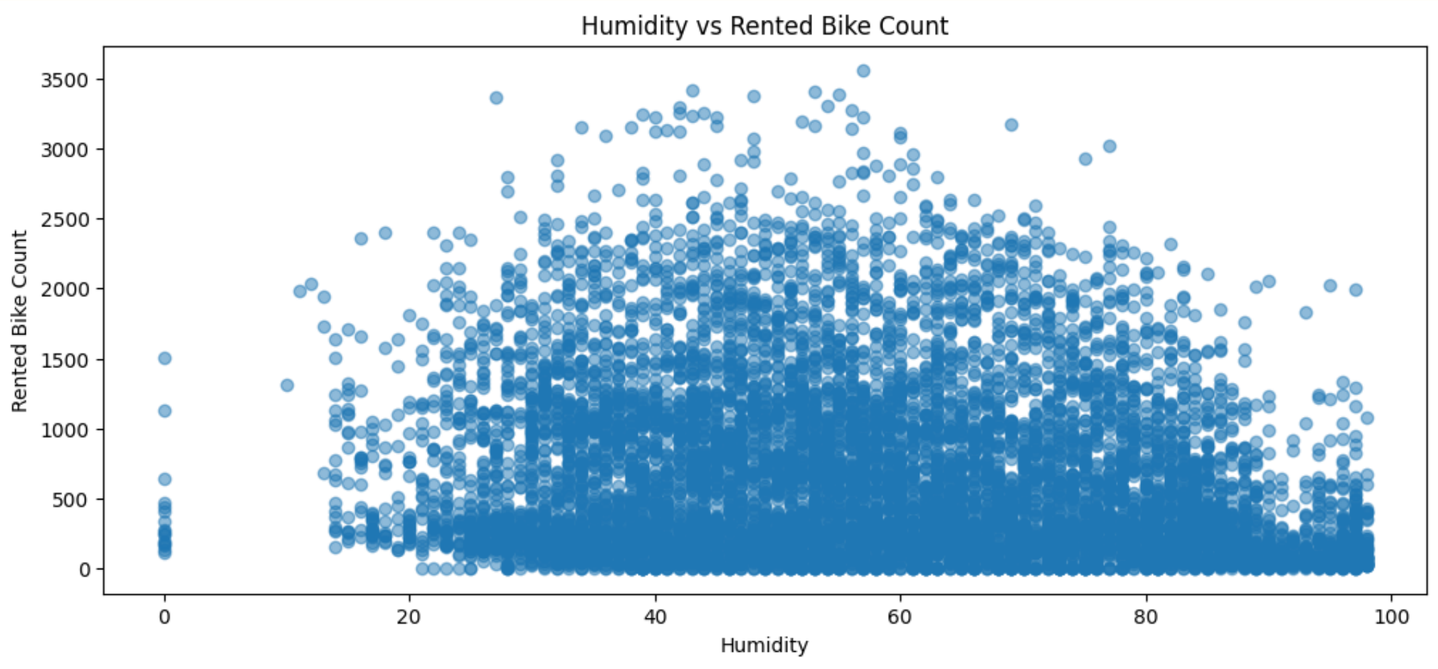
La demande de vélos décroît avec la baisse de la température.
Elle atteint son point culminant aux environs de 30 °C.
Pour des températures extrêmement élevées, on observe une diminution du nombre de vélos loués.

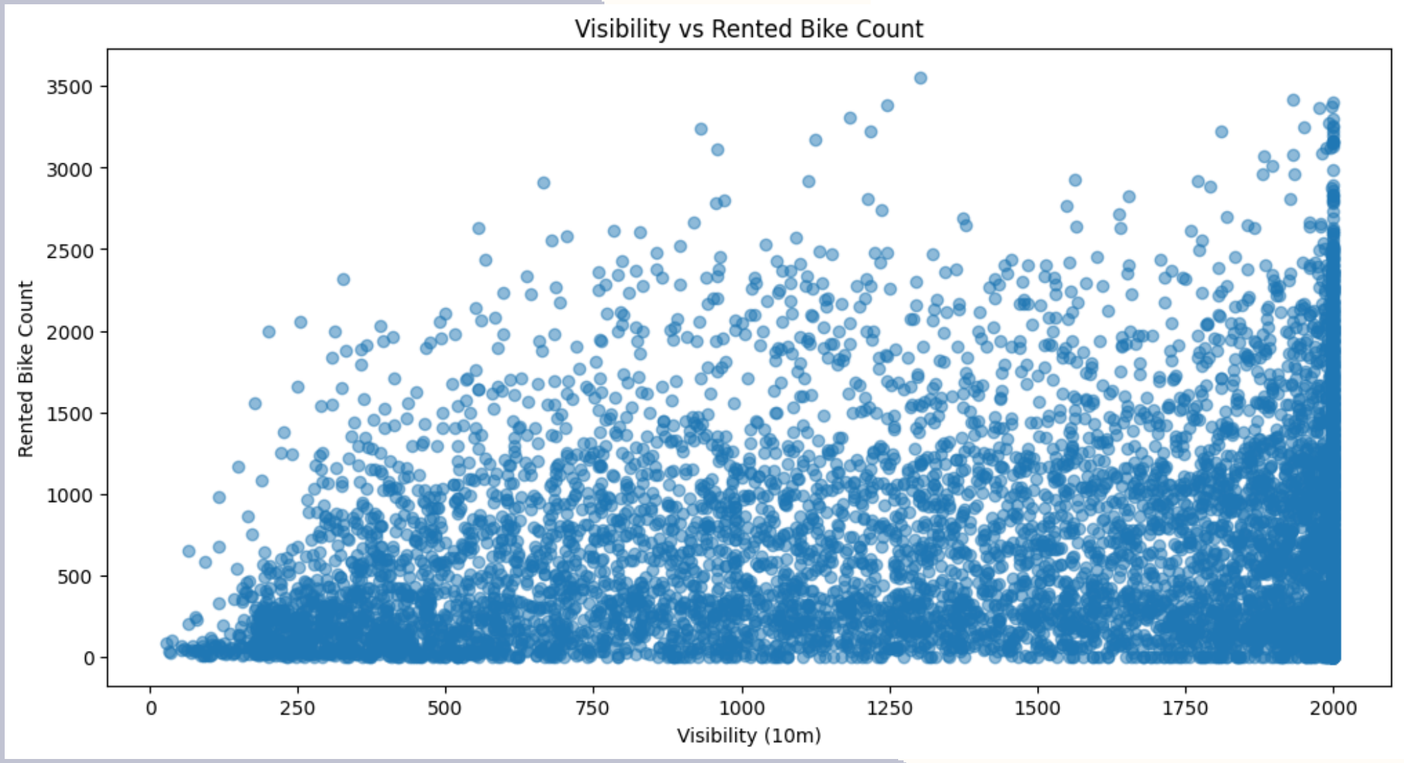


La demande de location de vélos atteint son point culminant entre 16h et 20h (à 18h). On observe également une hausse notable à 8h du matin.

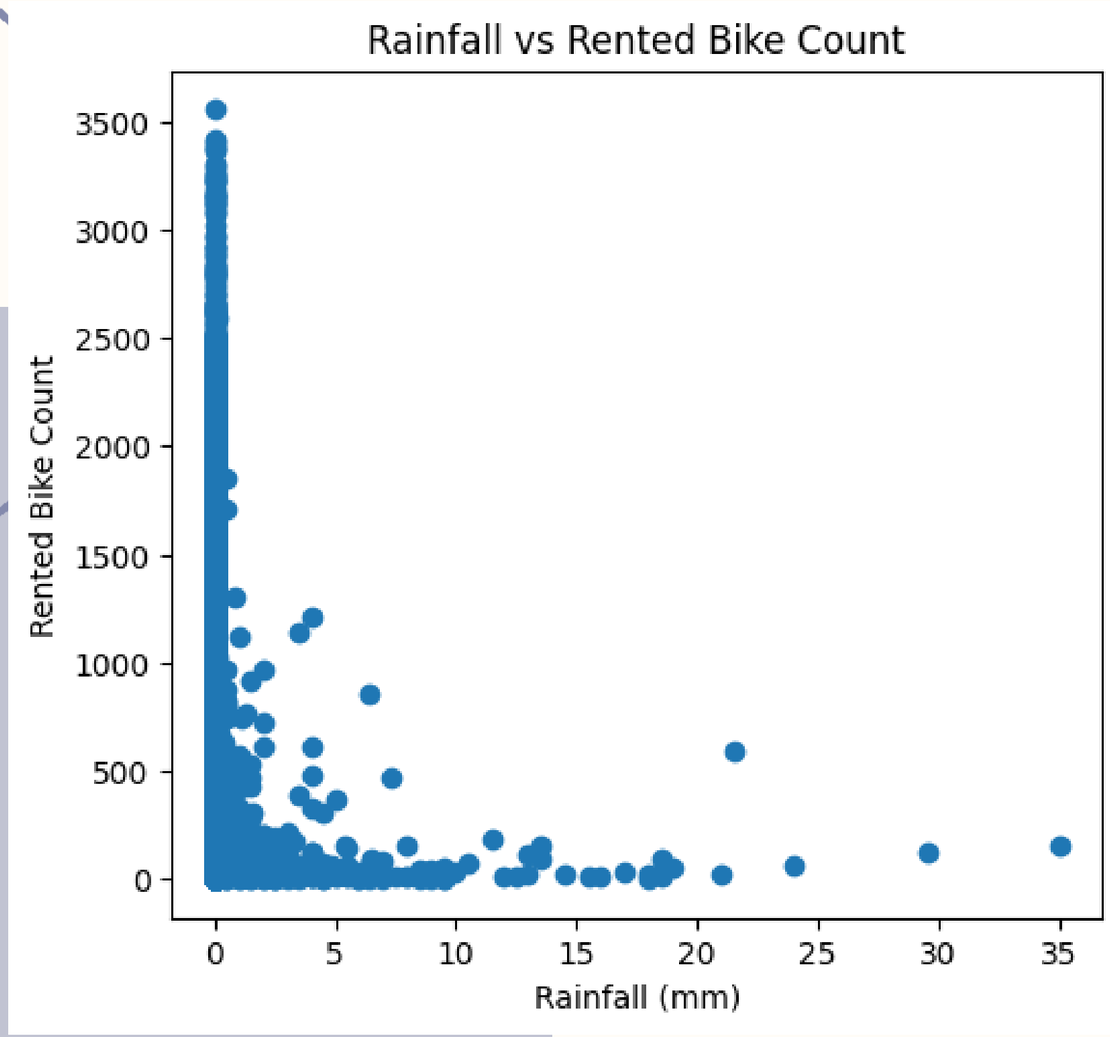
Cela suggère une utilisation fréquente pour les déplacements domicile-travail.

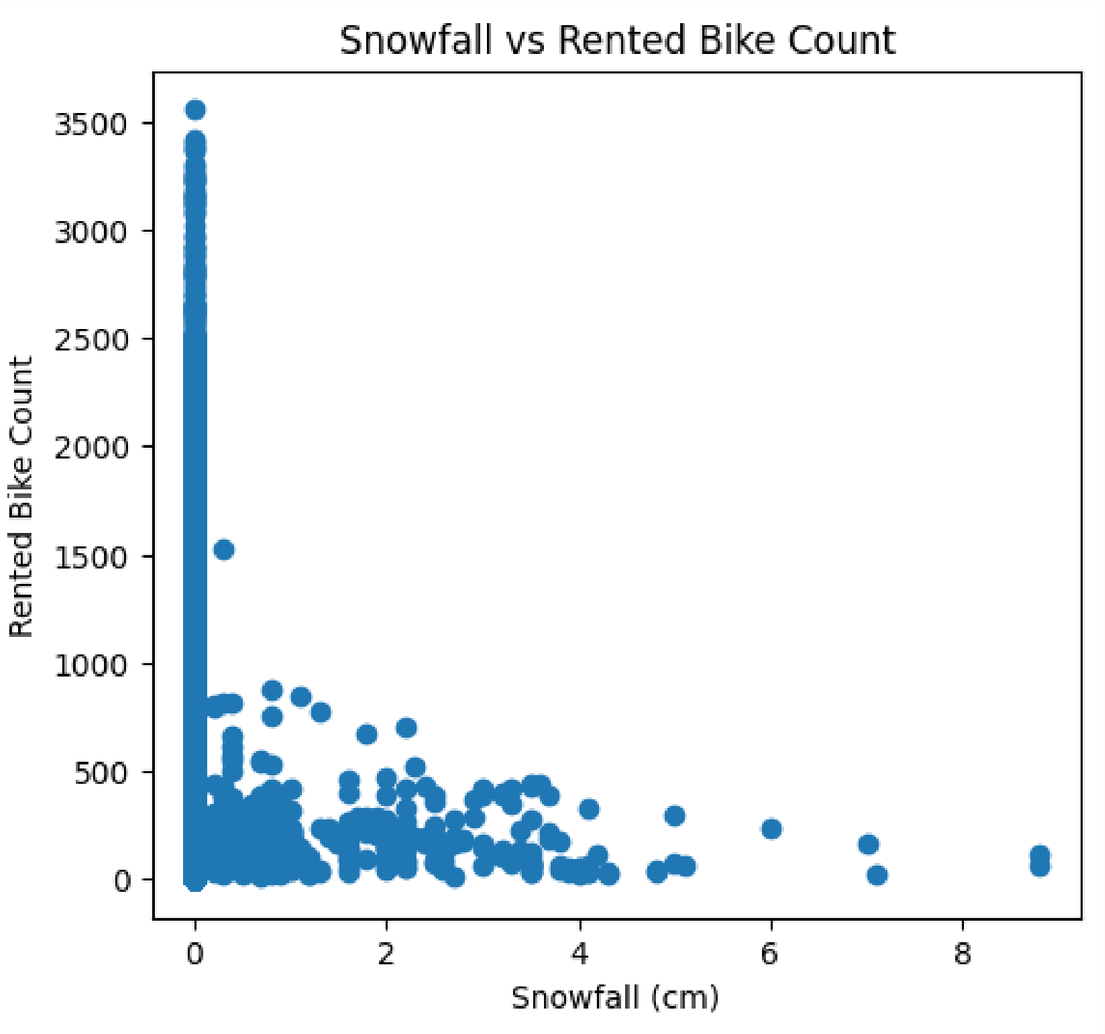
Pour des valeurs extrêmes d'humidité la demande de vélo commence à chuter





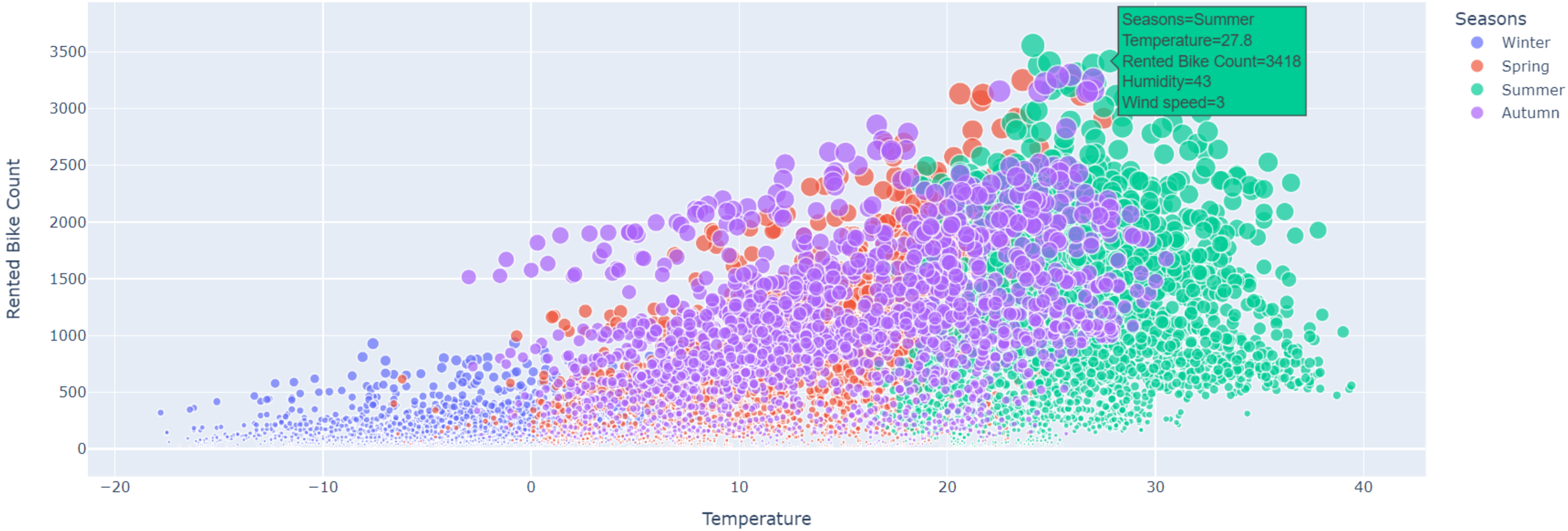
Tout naturellement, moins la visibilité est bonne, moins la demande de vélos est importante





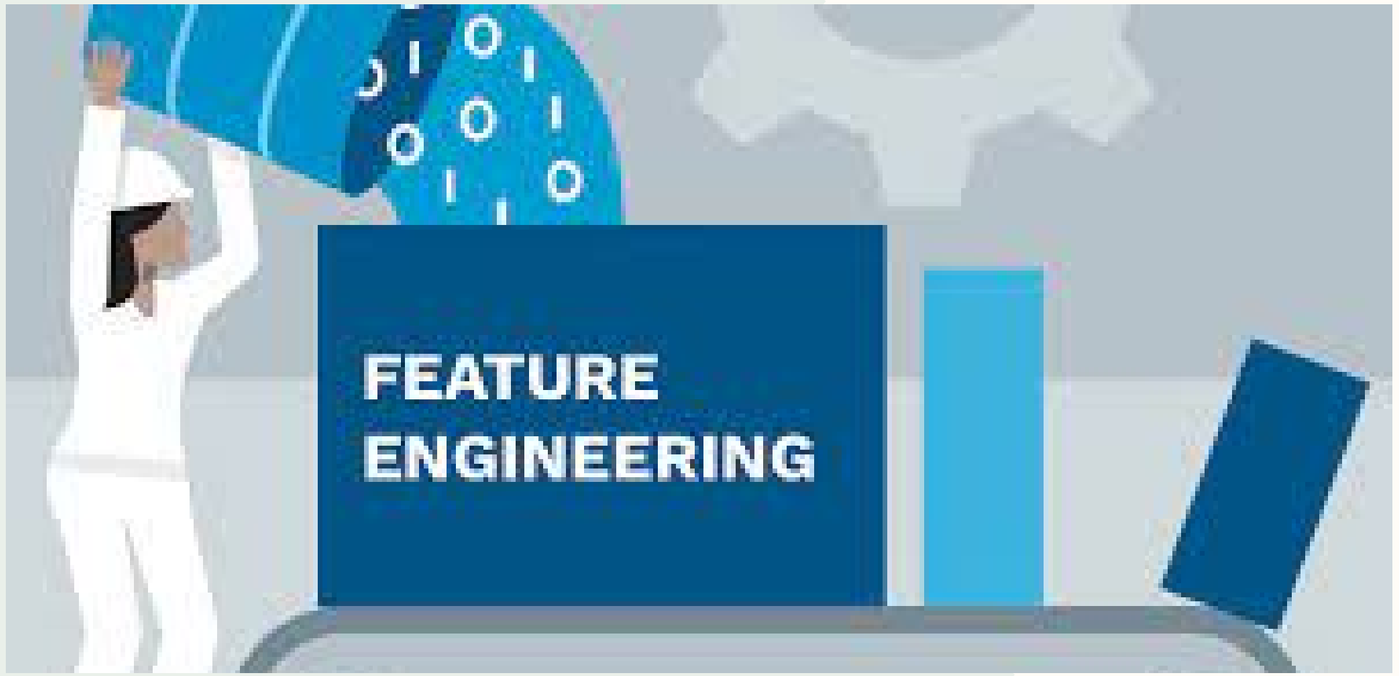
Les deux graphiques de dispersion présentent une tendance similaire, des niveaux de pluie et de neige plus élevés sont associés à une diminution des locations de vélos.

Résumé des corrélations entre la Température, le Nombre de Vélos Loués et les Conditions Météorologiques Saisonnieres



On retrouve les résultats déjà évoqués

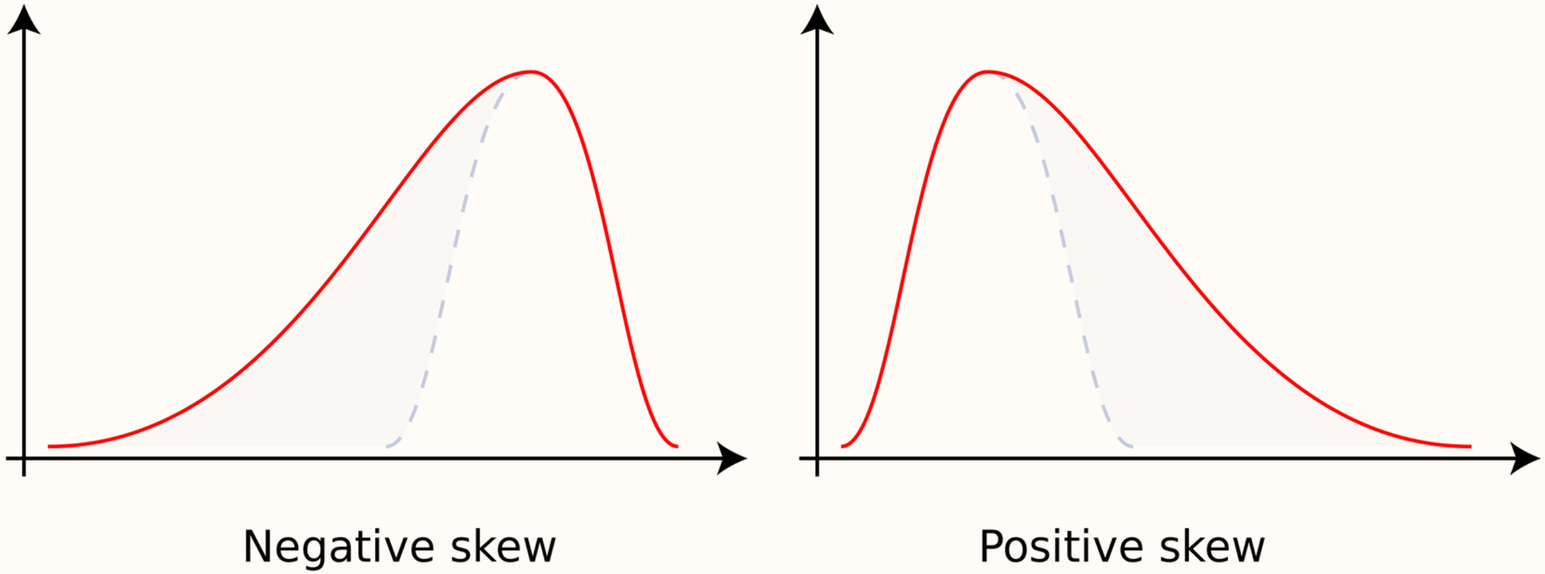
Feature engineering



Réduction de la skewness

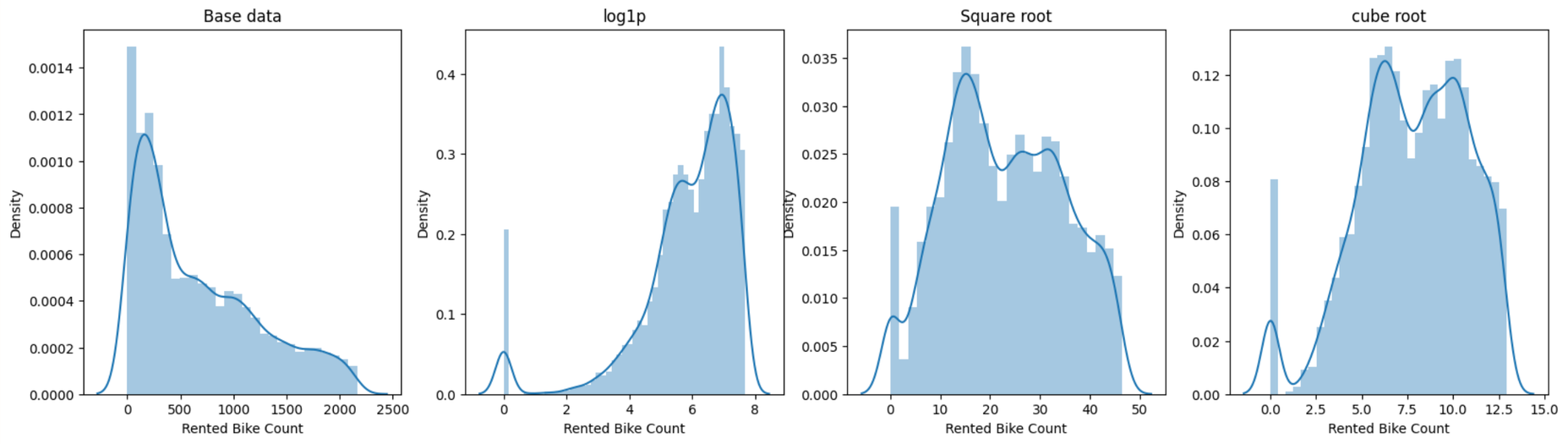
Skewness = coefficient d'asymétrie (skewness en anglais)
correspond à une mesure de l'asymétrie de la distribution d'une variable aléatoire réelle

Comment la réduire ?
Transformations ! (log, square root, cube root, inverse, etc...)



Application

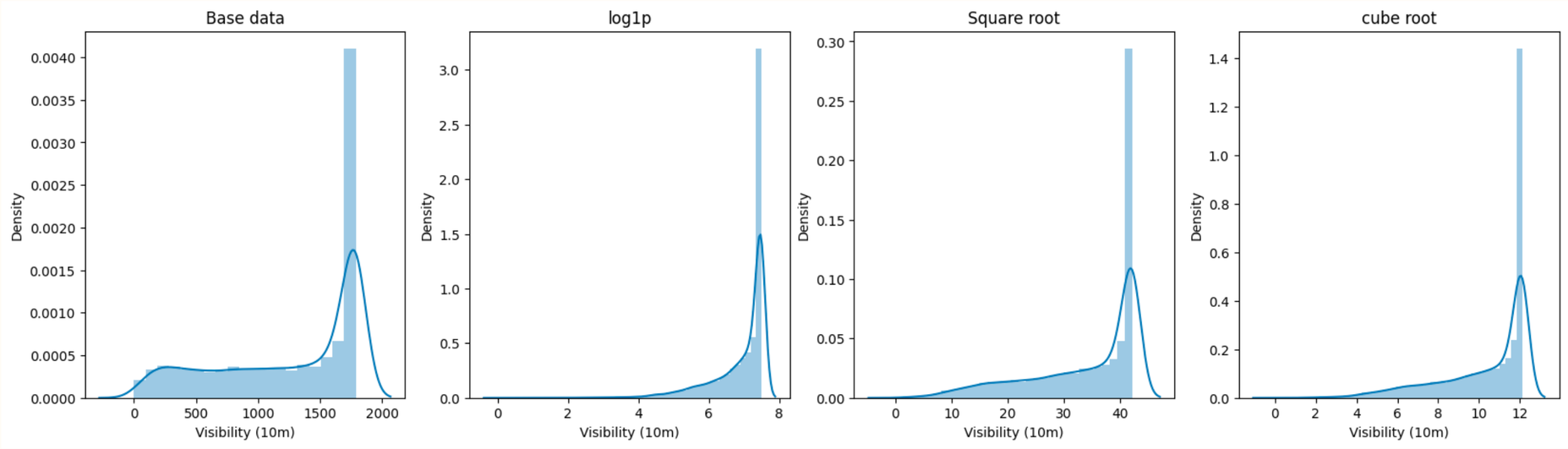
Sur la target: ‘Rented Bike Count’



Débat entre square root et cube root, mais dans les deux cas bien meilleurs que la donnée de base !

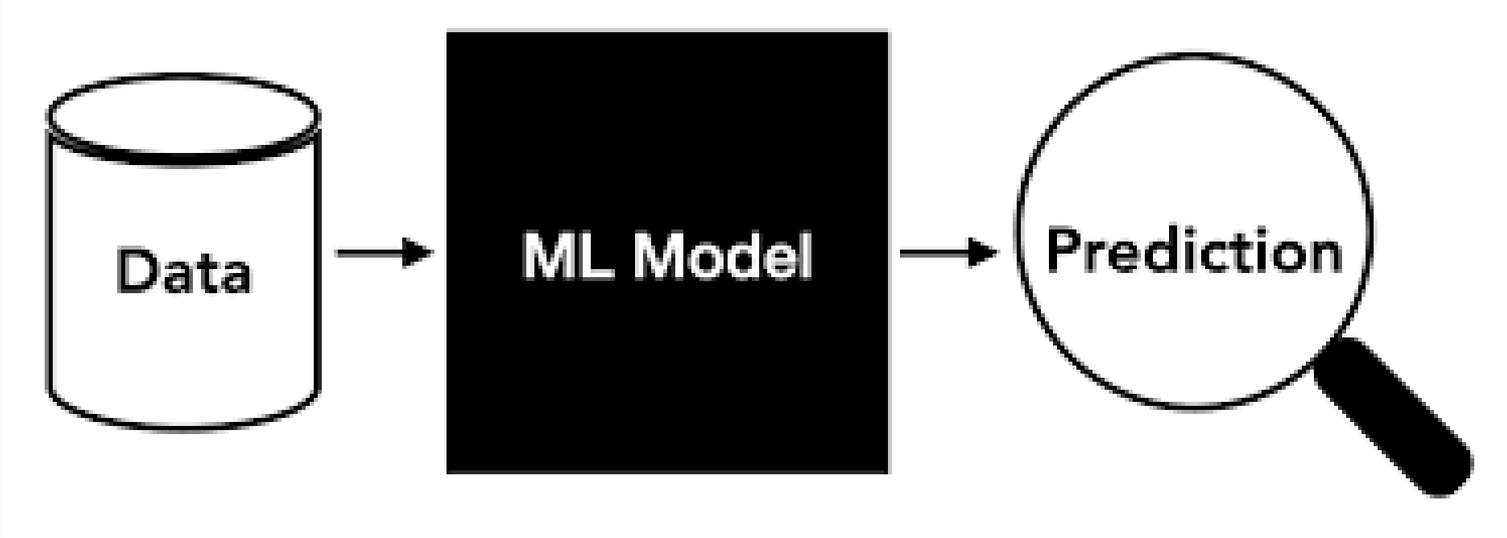
Moins de skewness => meilleures prédictions

Visibility



Malheureusement non applicable dans ce cas la,
pas de solution magique

Modeling



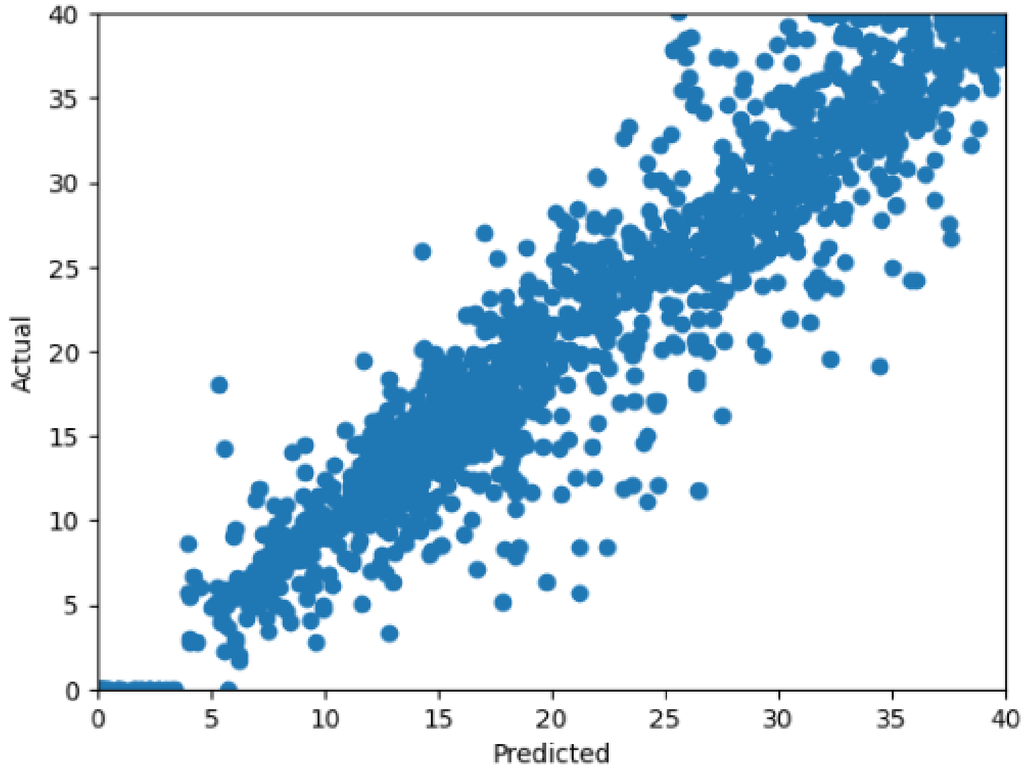
14 modèles testés :

- LGBM SS: LightGBM avec Standard Scaler
- LGBM: LightGBM
- LGBM MMS: LightGBM avec MinMax Scaler
- LGBM RS: LightGBM avec Robust Scaler
- Extra Trees: Extra Trees
- Random Forest: Random Forest
- Gradient Boosting: Gradient Boosting
- Decision Tree: Arbre de Décision
- SVR SS: Support Vector Regression avec Standard Scaler
- Linear Regression: Régression Linéaire
- Ridge SS: Régression Ridge avec Standard Scaler
- Lasso SS: Régression Lasso avec Standard Scaler
- KNN: K-Nearest Neighbors
- XGBoost

Exemples de 4 modèles testés sans GridSearch

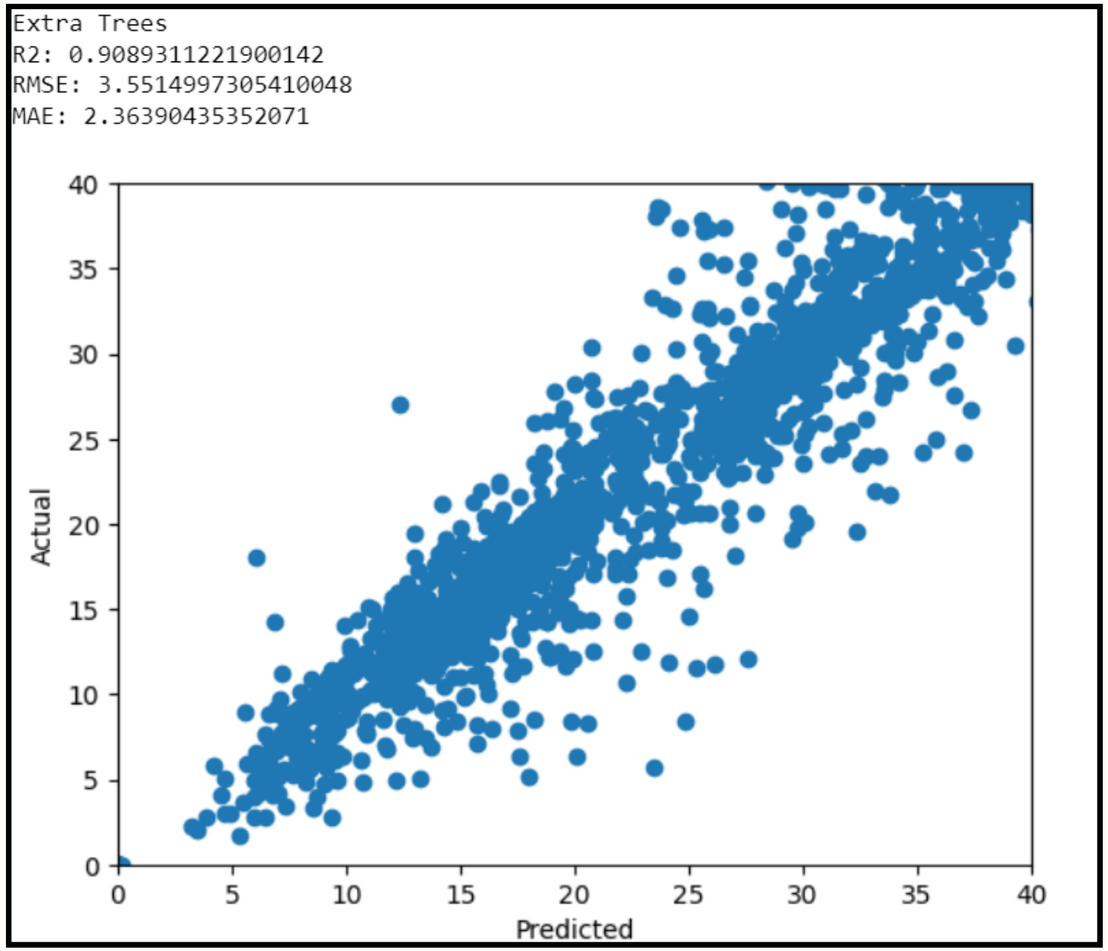
LGMB

LGBM
R2: 0.9138108817968883
RMSE: 3.455039463688245
MAE: 2.3905718772573397



Extra Trees Regressor

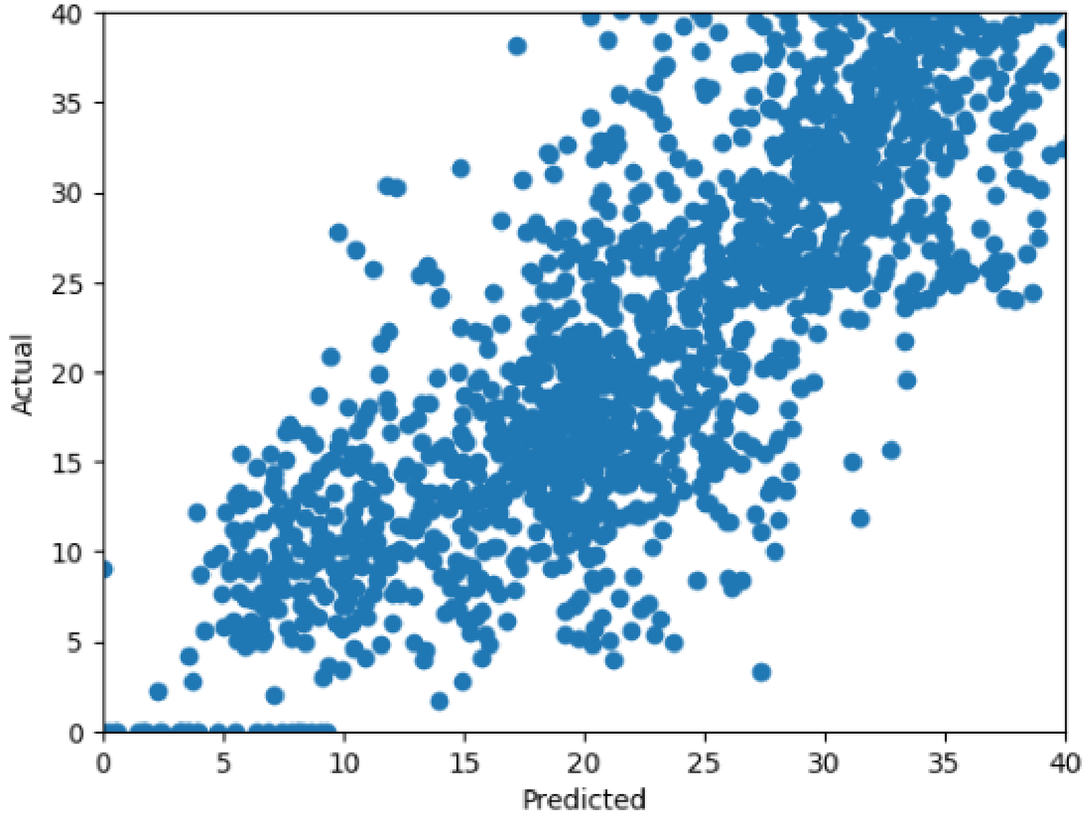
Extra Trees
R2: 0.9089311221900142
RMSE: 3.5514997305410048
MAE: 2.36390435352071



A noter que tous les LGMB (MMS, RS) sont plus performants que les autres modèles mais pour diversifier les visuels nous projetons le plus performant des LGMB .

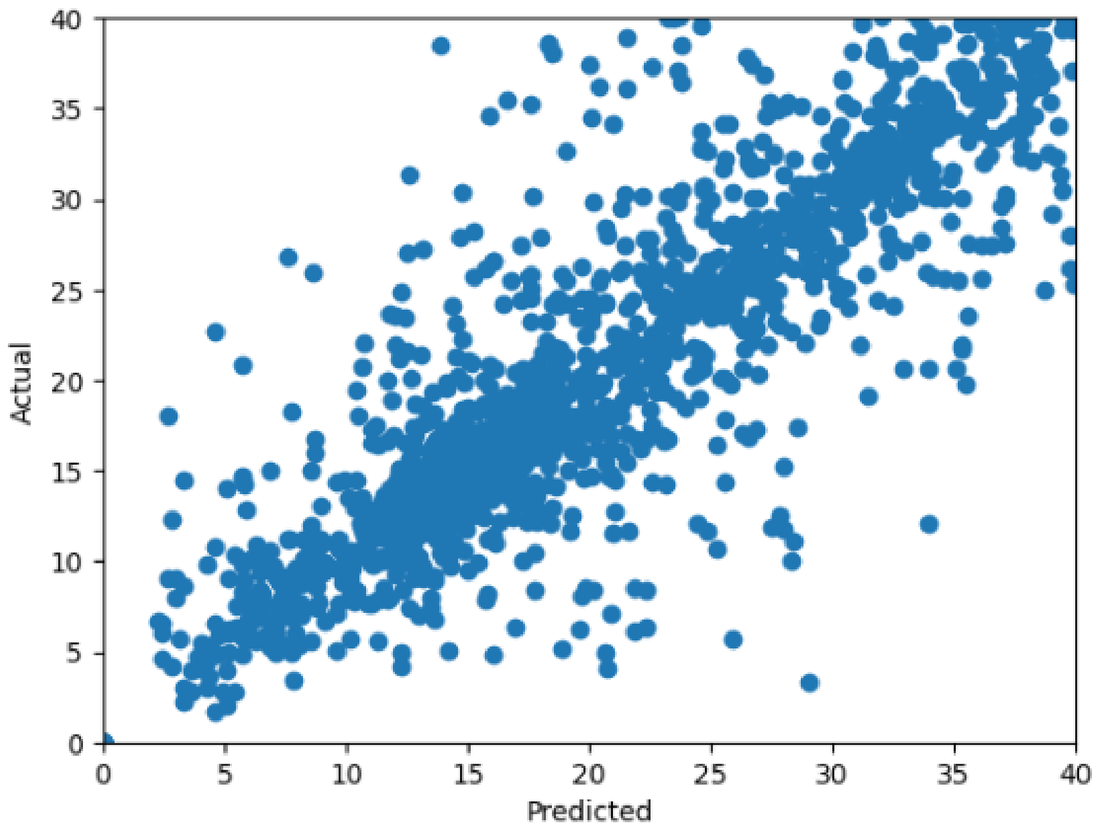
Linear Regression

```
Linear Regression
R2: 0.6729537203902765
RMSE: 6.730253151597643
MAE: 5.193499751576661
```



Decision Tree

```
Decision Tree
R2: 0.8074780798973031
RMSE: 5.163769948622941
MAE: 3.2627379371897507
```



Ces 2 modèles quant à eux font parties des modeles qui n'ont pas eu de très bon résultat

Nous avons effectué des GridSearch sur les modèles qui étaient les plus performants. Ce qui nous a permi d'avoir de meilleurs hyperparamètres :

lgbm_params = {'colsample_bytree': 1.0, 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 200, 'subsample': 0.8}

rf_params = {'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}

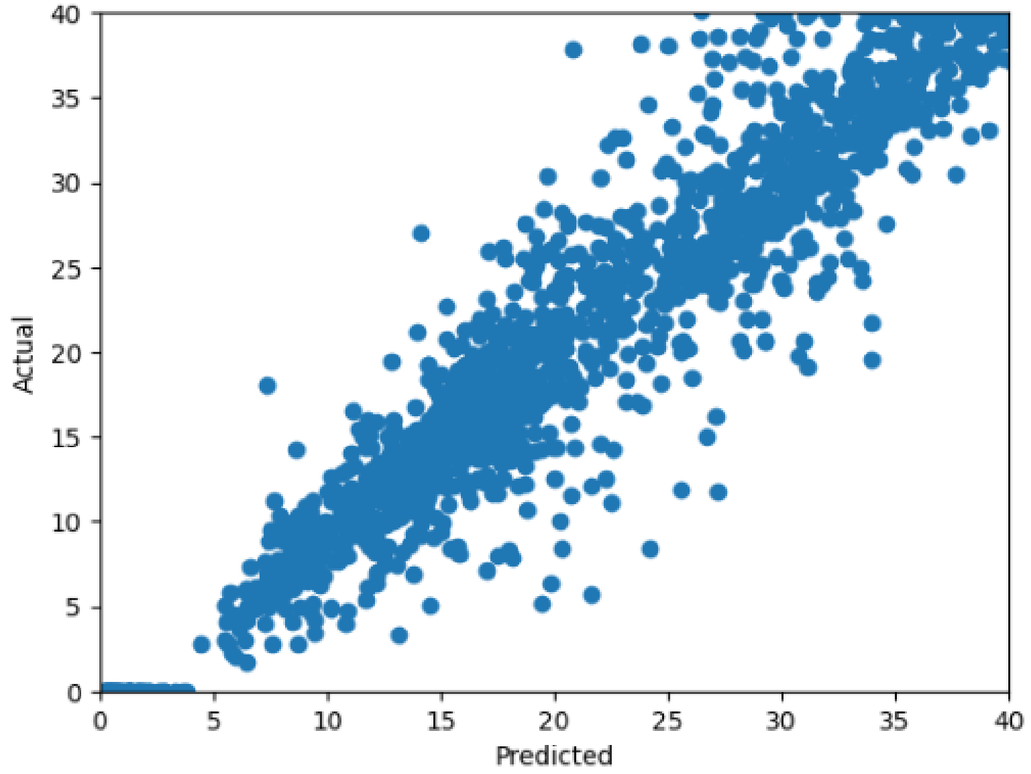
grad_params = {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 200, 'subsample': 1.0}

Best Hyperparameters for XGBoost with R2: {'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 200, 'subsample': 0.8}

4 modèles testés après avoir effectué un GridSearch

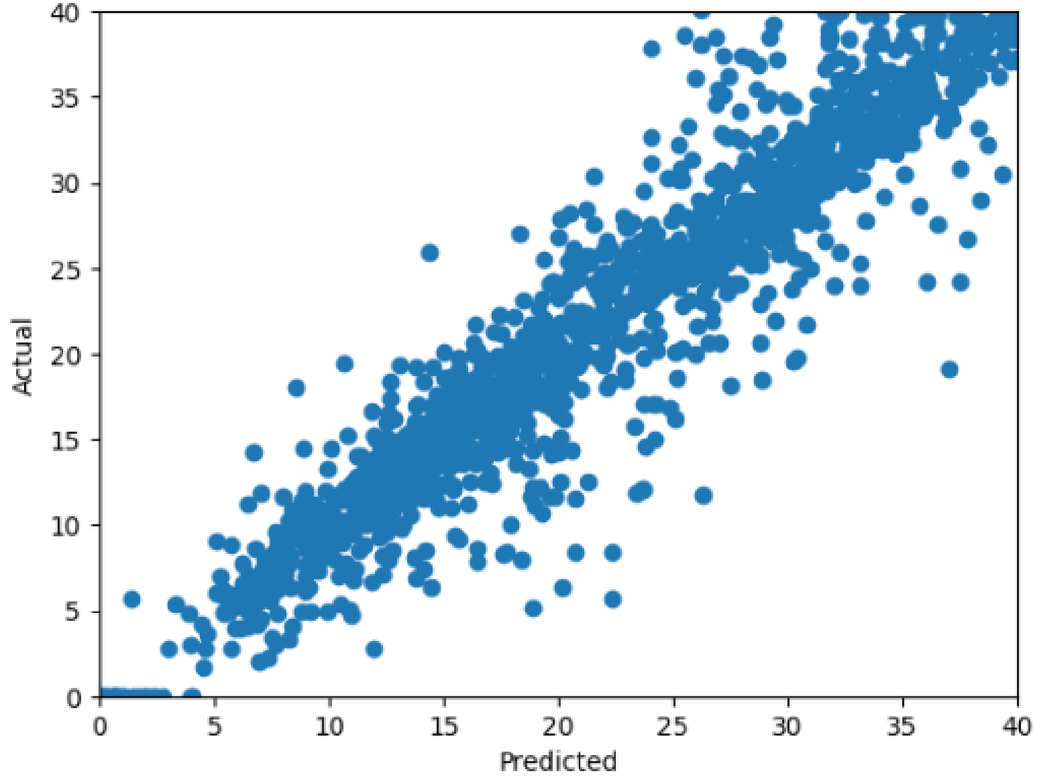
Random Forest

```
Random Forest
R2: 0.9017586533387211
RMSE: 3.6887051922324017
MAE: 2.5545983831444206
```



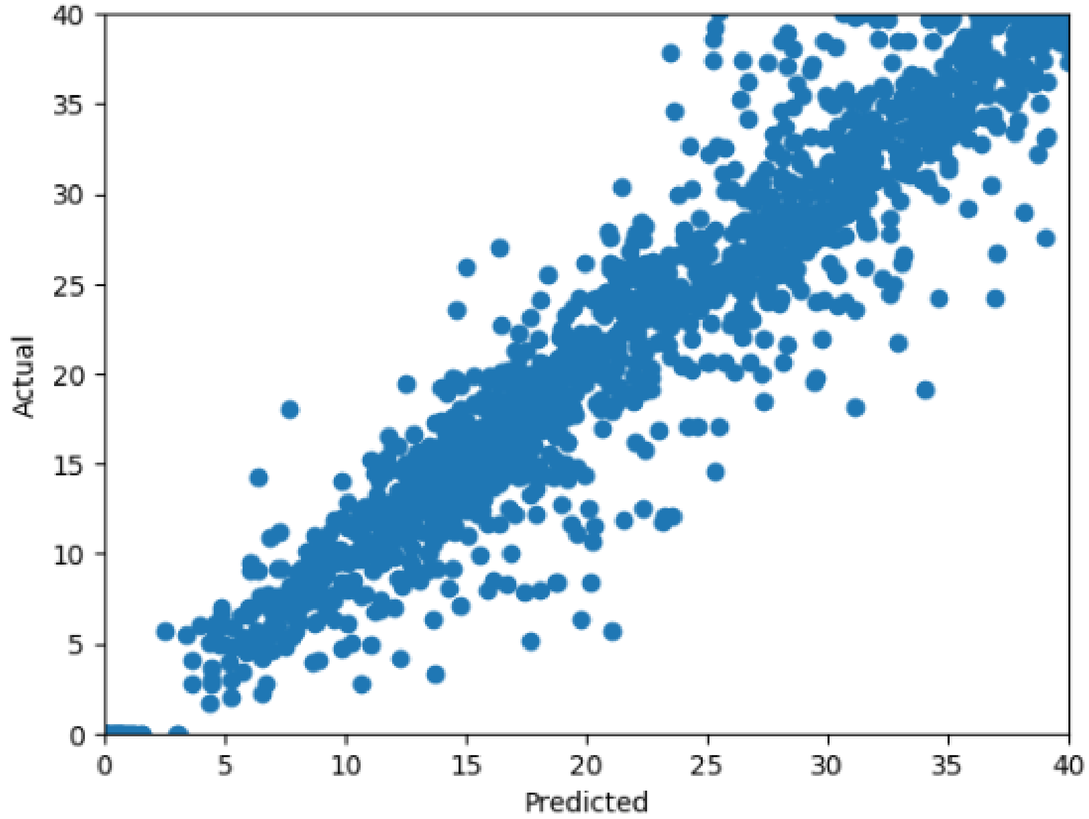
LGMB SS

```
LGBM SS
R2: 0.9220028433589977
RMSE: 3.286746271938733
MAE: 2.198098674095139
```



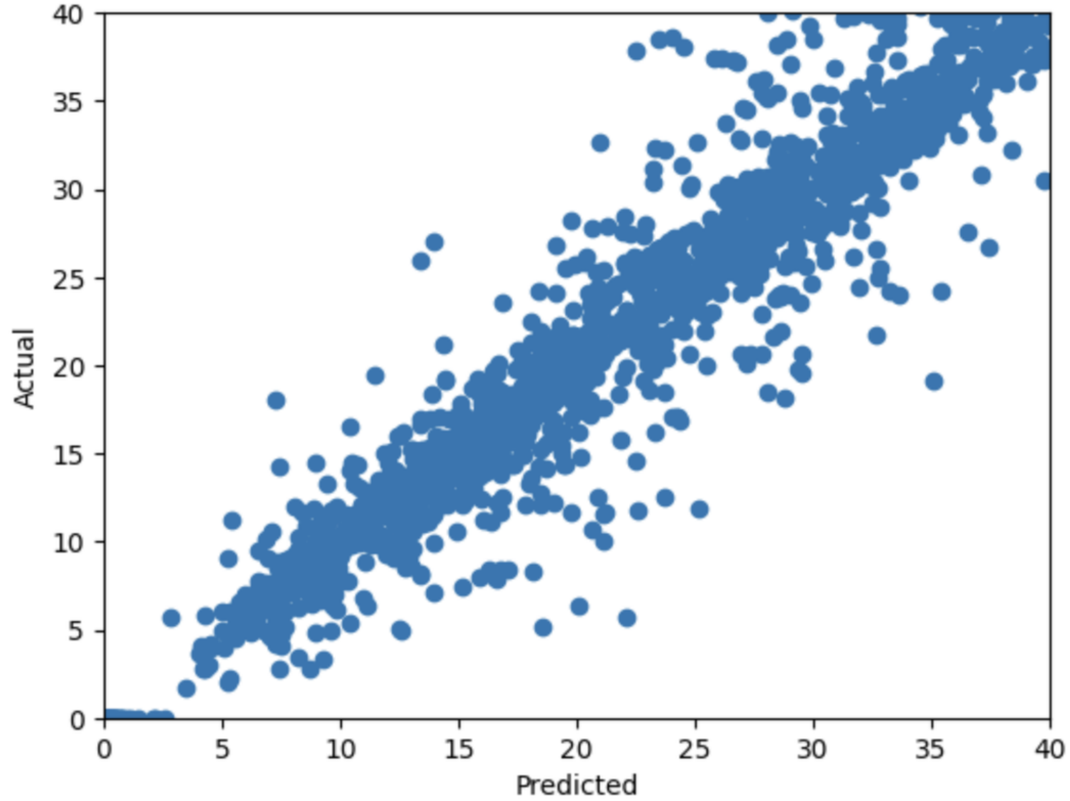
Gradiant Boost

Gradient Boosting
R2: 0.921885454258396
RMSE: 3.289218689796545
MAE: 2.1767147687983344



XGB RS

XGBoost GSCV RS
R2: 0.9273153767161113
RMSE: 3.1728392214165715
MAE: 2.0313328412217944



Résumé des résultats obtenus (ordre des plus performants

Model	R2	RMSE	MAE
XGBoost GSCV RS	0.927315	3.172839	2.031333
XGBoost GSCV MMS	0.927315	3.172839	2.031333
XGBoost GSCV SS	0.927315	3.172839	2.031333
Gradient Boosting GSCV	0.923667	3.251496	2.108596
LGBM GSCV MMS	0.922641	3.273279	2.190377
LGBM GSCV RS	0.922104	3.284610	2.197318
LGBM GSCV SS	0.922003	3.286746	2.198099
XGBoost	0.921515	3.297006	2.210608
LGBM	0.920220	3.324107	2.269130
Random Forest GSCV	0.907454	3.580178	2.451091
Extra Trees GSCV SS	0.902350	3.677585	2.487381
Decision Tree	0.819277	5.003040	3.086179
SVR SS	0.779140	5.530769	3.900389
Linear Regression	0.673089	6.728865	5.193986
Ridge SS	0.673071	6.729046	5.194290
Lasso SS	0.636850	7.092021	5.615643
KNN	0.551247	7.883706	5.566782

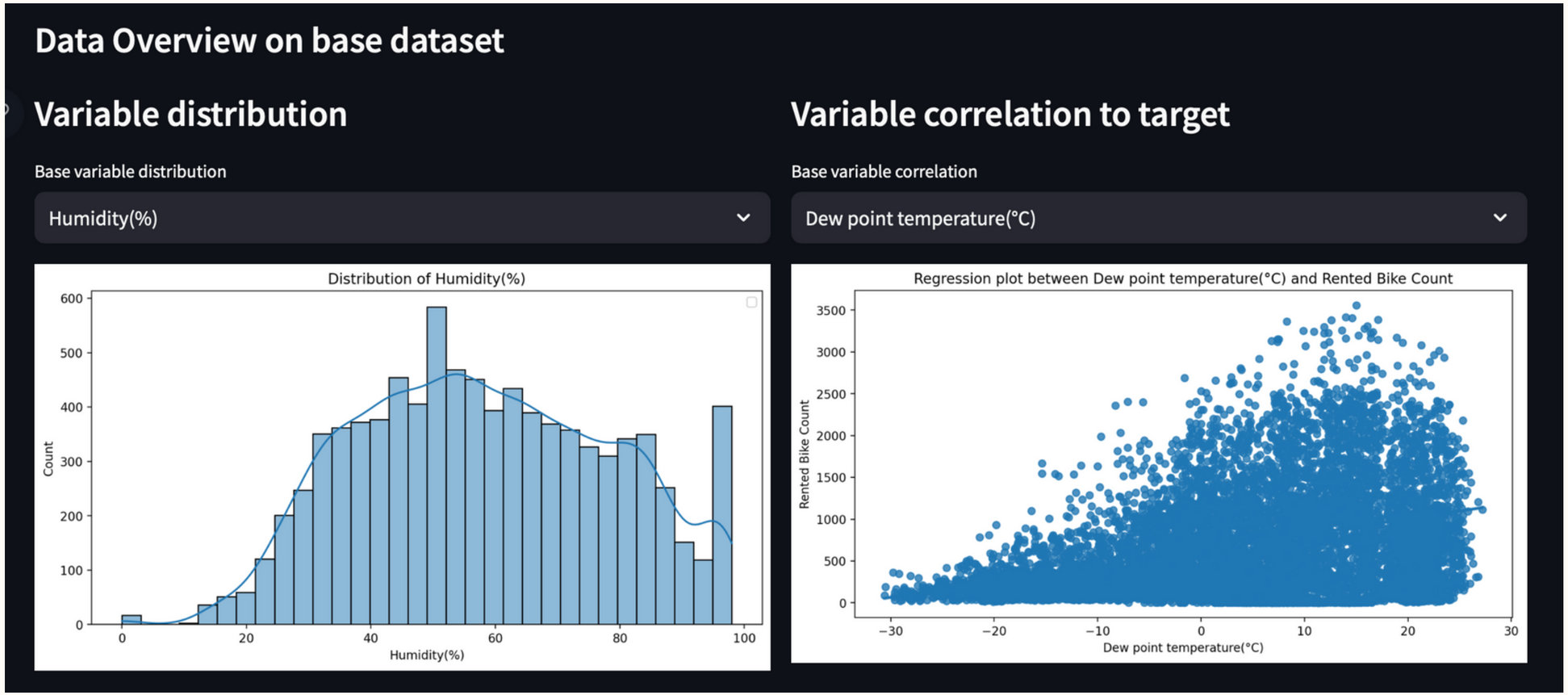
XGboost après GridSearch et un RobustScaler est donc le modèle le plus performant

Visualisations interactives

Visualisations interactives sur le dataset

Distribution des variables

Corrélation (avec la target)

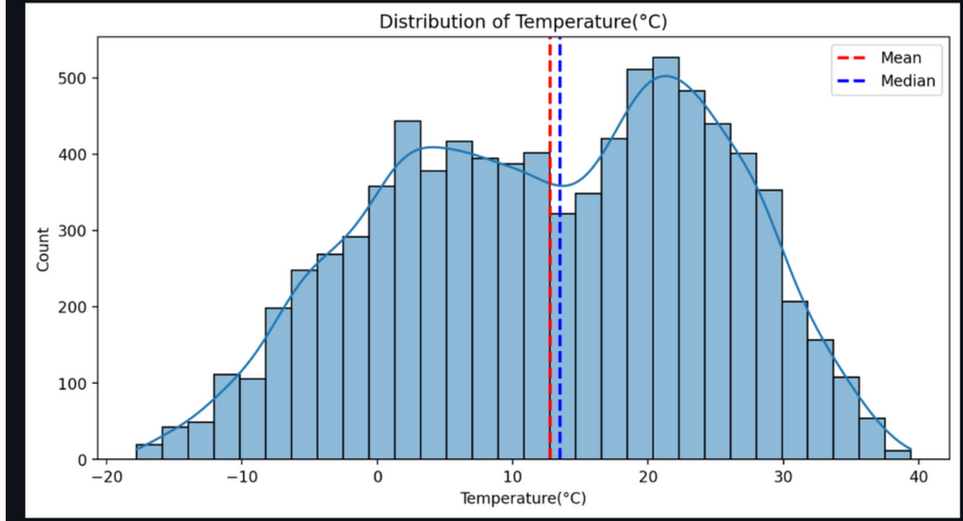


Data Overview on filtered dataset

Numerical variables

Numerical base distribution

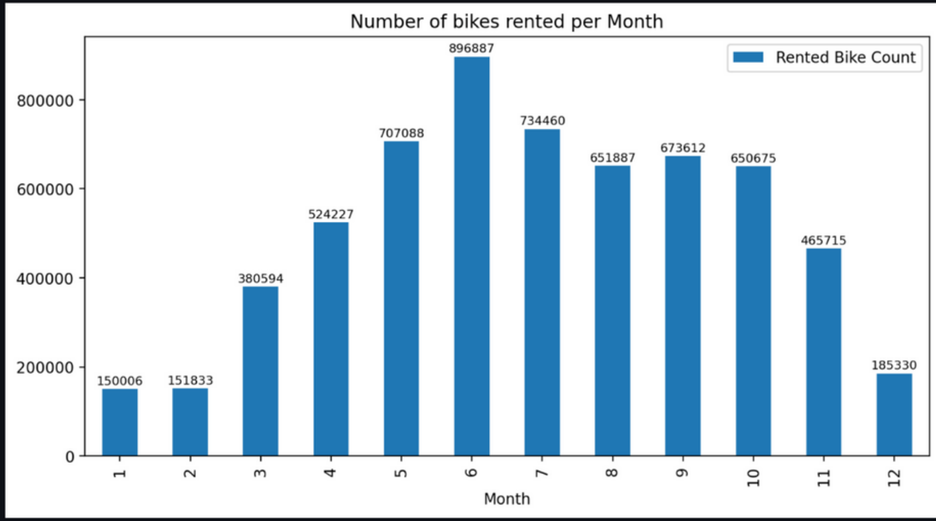
Temperature(°C)



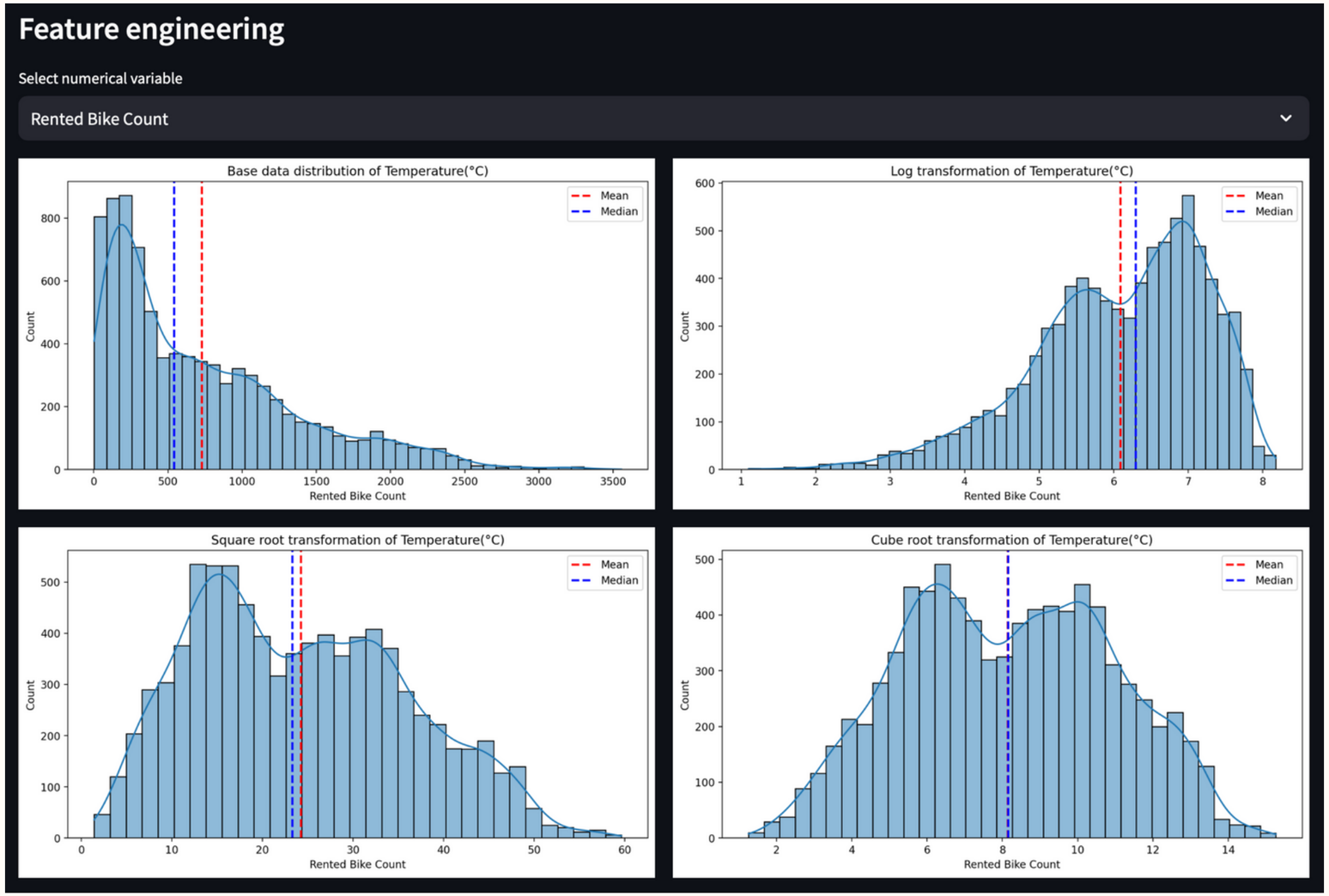
Discrete variables

Discrete variable distribution

Month



Sur le dataset filtré



Comparaison des distributions après transformations

Modeling interatrif

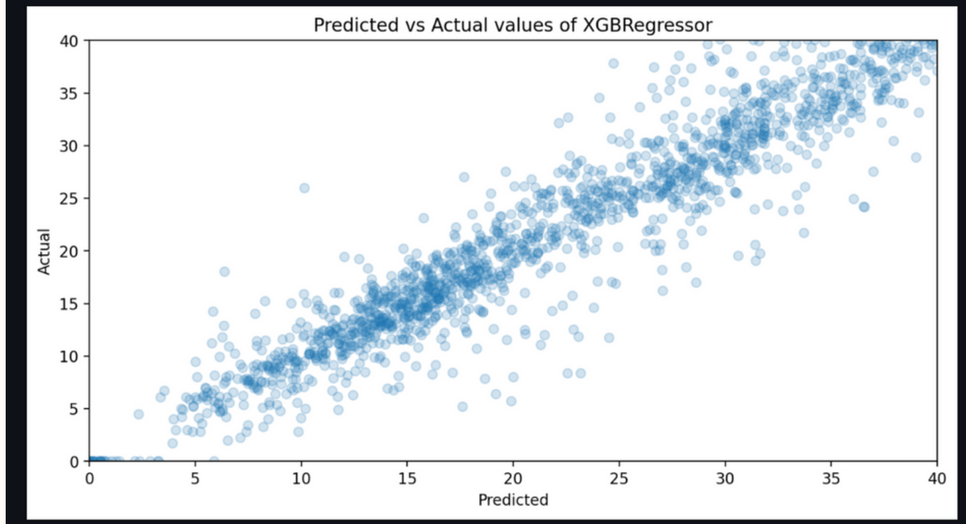
Modeling

Select model

XGBRegressor

Select scaler

StandardScaler



R2 score: 0.919

MSE: 11.208

MAE: 2.292

Model description: XGBoost Regressor is an implementation of gradient boosted decision trees designed for speed and performance.

Model key feature: Optimized for distributed computing and improved regularization to prevent overfitting.

MERCI

DE VOTRE ATTENTION