Test technique

Héloïse THERO

11 juillet 2019

Partie I :
Statistiques descriptives

Description du jeu de données

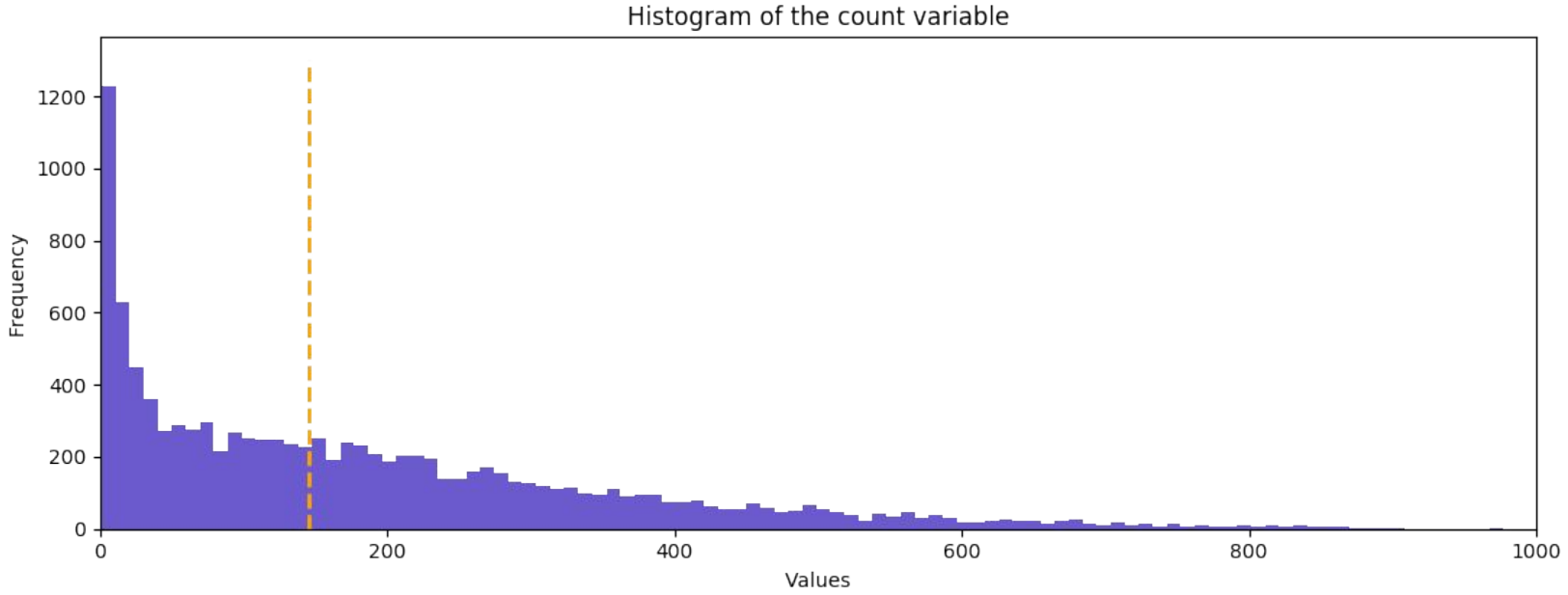
- Les données consistent en une table de données appelée “bike_data.csv”, qui contient **10 886 lignes** et **10 colonnes.** Il n’y a **pas de valeur manquante** dans la table (pas de NaN).
- Voici une brève description des différentes colonnes :

Nom	**datetime**	**season**	**holiday**	**workingday**	**weather**	**temp**	**atemp**	**humidity**	**windspeed**	**count**
Type de variable		catégorielle cyclique	catégorielle	catégorielle	catégorielle ordonnée	quantitative continue	quantitative continue	quantitative discrète	quantitative continue	quantitative discrète
Médiane	-	3	0	1	1	20.500	24.240	62	12.998	145
Minimum	'2011-01-01 00:00:00'	1	0	0	1	0.82	0.760	0	0	1
Maximum	'2012-12-19 23:00:00'	4	1	1	4	41.00	45.455	100	56.9969	977

- Il est à noter que **des lignes sont manquantes** puisqu’en prenant la datetime minimum et maximum et en comptant 24 lignes par jour, on devrait obtenir 17 232 lignes. Or la table n’en contient que 10 886.

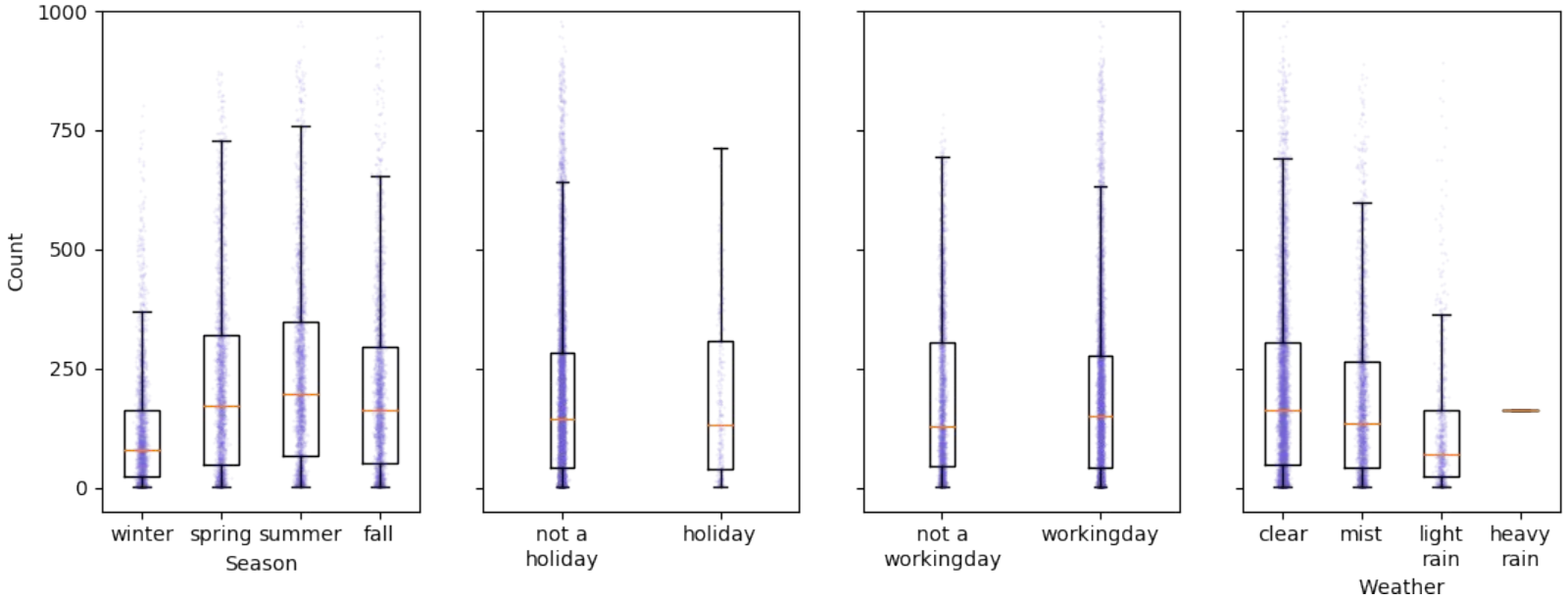
La variable à prédire

- Regardons plus précisément la colonne “count”, puisqu’il s’agit de la variable à prédire.
- On peut voir qu’elle contient de **nombreuses valeurs faibles**. D’ailleurs sa médiane est à 145, pour des valeurs allant de 1 à 977.
- Sa distribution ressemble à celle d’une loi géométrique mais avec une longue traîne.



Prédicteurs catégoriels

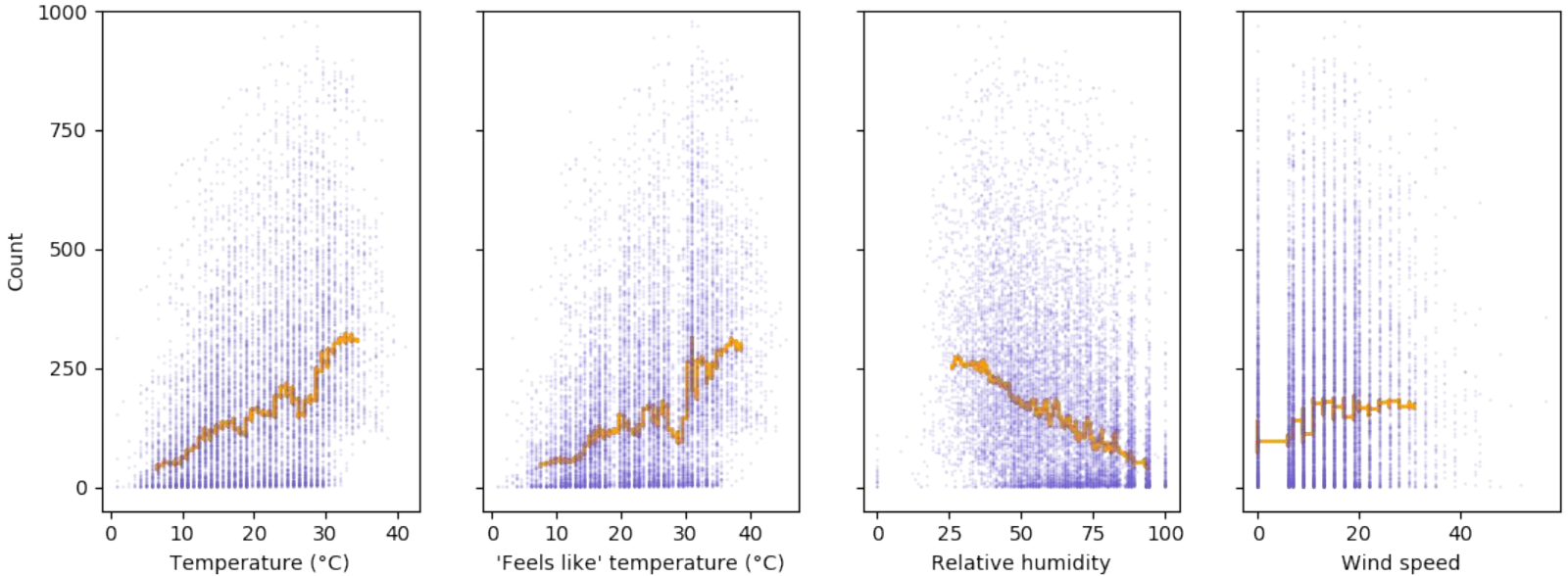
- Les prédicteurs catégoriels qui semblent influencer la demande en vélos sont :
  - **la saison**, avec le printemps et l'été correspondant à des périodes de forte demande,
  - **la météo**, avec un temps clair correspondant à des périodes de forte demande.
- Chaque point violet correspond à un point de donnée, avec une boîte à moustache en noir et orange.



NB : Contrairement à ce qui est indiqué dans la description des données, j'ai plutôt encodé la saison ainsi : 1 = winter, 2 = spring, 3 = summer, 4 = fall pour que les saisons soient cohérentes avec les datetimes.

Prédicteurs quantitatifs (1/2)

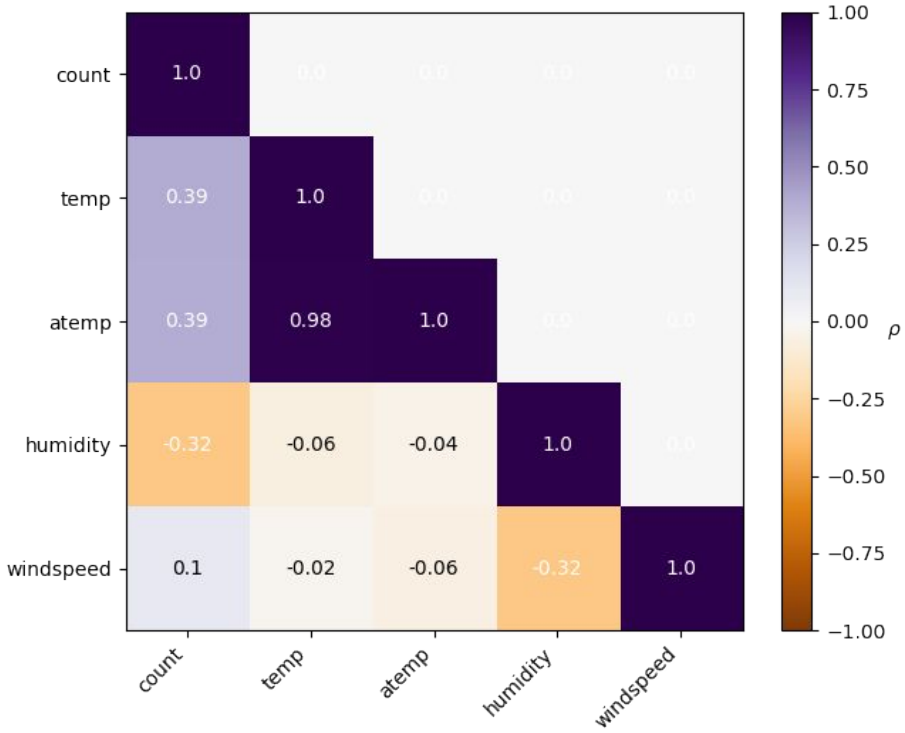
- Les prédicteurs quantitatifs qui semblent influencer la demande en vélos sont :
  - **la température réelle** et **ressentie**, qui corrèle positivement avec la demande,
  - **l'humidité**, qui corrèle négativement avec la demande.
- Chaque point violet correspond à un point de donnée, avec une médiane glissante en orange.



Prédicteurs quantitatifs (2/2)

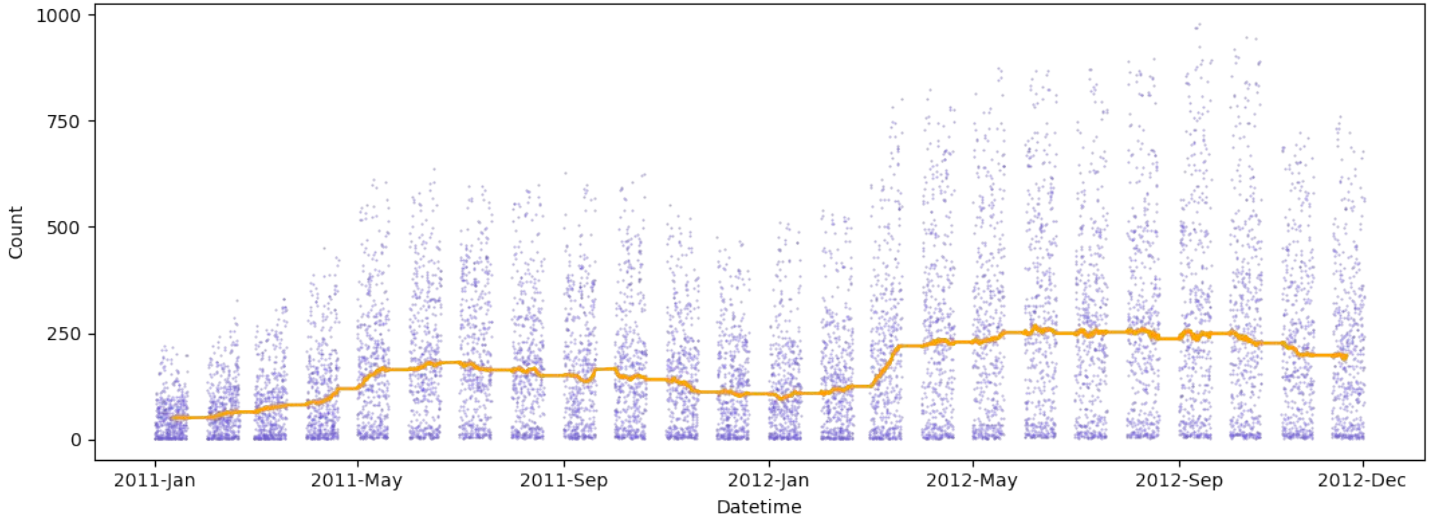
- J'ai ensuite visualisé la **matrice de corrélation** entre la variable à prédire et les différents prédicteurs quantitatifs. On constate à nouveau que :
  - temp et atemp corrèlent positivement avec count
  - humidity corrèle négativement avec count.

- Au niveau des corrélations entre les prédicteurs, on constate que :
  - **les températures réelle et ressentie sont fortement corrélées positivement** (ce qui était plutôt prévisible mais peut-être pas à ce point).
  - l'humidité et la force du vent sont faiblement corrélées négativement.



Prédicteurs extraits du datetime (1/2)

- J'ai également visualisé la demande en vélo selon la date. On remarque une certaine **saisonnalité** dans la demande ainsi qu'une **hausse progressive**.
- On remarque que les données ne contiennent que les **19 premiers jours de chaque mois** (en effet l'objectif de la compétition kaggle était de prédire les derniers jours du mois).

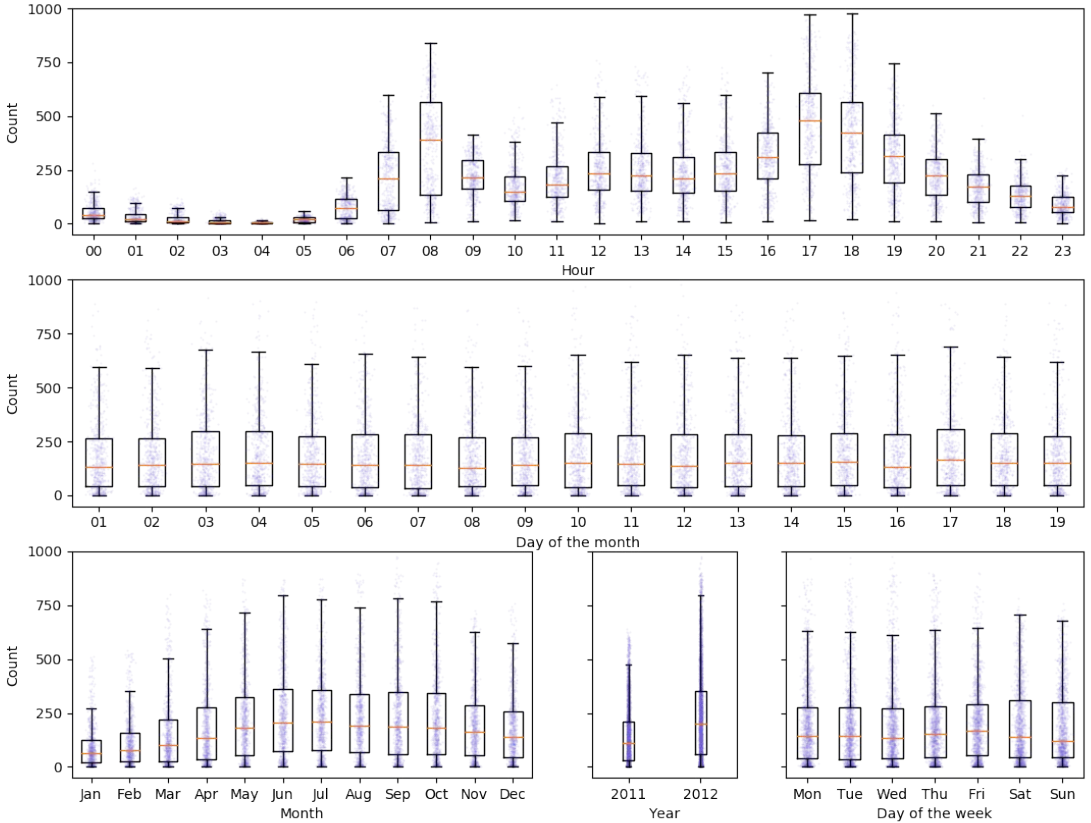


NB : Même en ne comptabilisant que 19 jours par mois, on devrait obtenir 19 x 12 x 24 x 2 = 10 994 lignes et non 10 886 comme le fichier actuel. Il reste donc certaines lignes manquantes mais cela représente moins de 1% des données donc c'est un manque d'information négligeable.

Prédicteurs extraits du datetime (2/2)

- J'ai ensuite ensuite **extrait différentes variables** de la colonne "datetime" : l'heure, le jour du mois, le mois, l'année et le jour de la semaine.

- Les prédicteurs qui semblent influencer la demande en vélo sont :
  - **l'heure** avec deux pics à 8h et à 17-18h
  - **le mois** avec une augmentation en juin-juillet
  - **l'année** avec une augmentation en 2012 par rapport à 2011

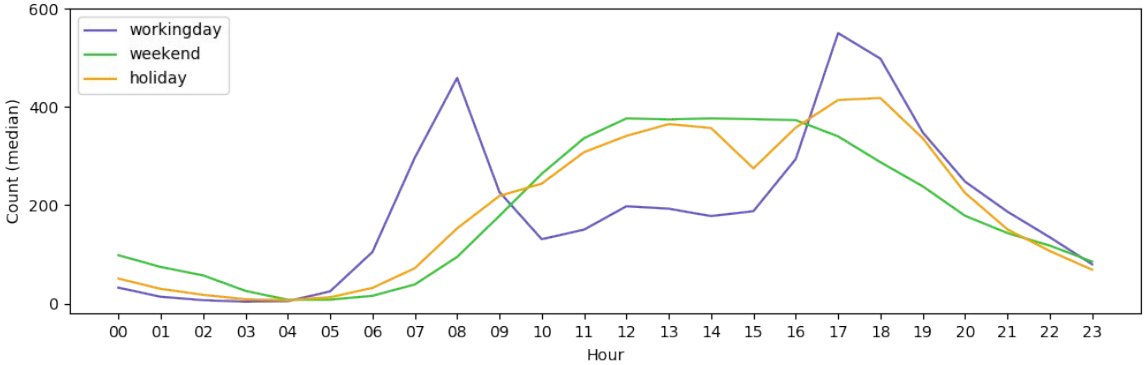


Interaction entre l'heure et le jour

- J'ai finalement **créé une variable** appelée "specialday" qui correspond à 0 pour les jours de semaine travaillés, à 1 pour les jours de weekend et à 2 pour les jours fériés. Cette variable résume donc les variables workingday et holiday :

workingday	holiday	specialday
1	0	0
0	0	1
0	1	2

- J'ai été inspirée par cette étude qui montre que l'effet de l'heure dépend du jour de la semaine. J'ai alors voulu regarder l'effet de l'heure selon la variable "specialday" :
  - Le **pic de demande à 8h** n'apparaît **que pendant un jour travaillé**, et pas pendant un jour de week-end ou férié
  - Le **pic à 17-18h** est **plus grand les jours travaillés que fériés** et disparaît pendant les jours de week-end.



Partie II :
Machine Learning

Préparation des données

- **Les données ont été mélangées** pour ne plus avoir les datetime dans l'ordre (de janvier 2011 à décembre 2012). En effet nous avons vu que la demande en vélo a tendance à évoluer dans le temps. Un modèle entraîné sur l'année 2011 aura donc du mal à prédire les valeurs pour l'année 2012.

- Les variables catégorielles avec plus de deux valeurs ont été transformées avec un **one-hot encodeur.** Par exemple, voici comment la variable "weather" a été transformée :

weather	split_weather_1	split_weather_2	split_weather_3	split_weather_4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

- L'avantage du one-hot encodeur est de pouvoir capturer les **effets non-linéaires** de l'heure ou du mois par exemple.

Choix du modèle

- J'ai choisi d'utiliser un **modèle de régression linéaire**. En effet il s'agit d'un modèle simple à entraîner, et dont les paramètres sont très informatifs. Il s'agit généralement d'un premier modèle utile pour servir de benchmark. De plus un modèle linéaire avec des variables extraites du datetime et encodées en one-hot peut s'avérer très performant pour prédire une série temporelle.
- J'ai utilisé une **régularisation de Tikhonov** (*ridge*) avec alpha = 0.01 pour éviter le sur-apprentissage.

- J'ai entraîné ce modèle sur **deux jeux de données différents** :
  - un jeu de données avec les prédicteurs présents dans la table de données initiale
  - un jeu de données avec les prédicteurs initiaux, plus les features additionnelles présentées dans la partie I
- Voici donc la liste des prédicteurs pour chaque jeu de données :

Modèle sans features additionnelles	Modèle avec features additionnelles
season, weather, temp, atemp, humidity, windspeed	season, weather, temp, atemp, humidity, windspeed
holiday, workingday	specialday
-	year, month, day, hour, weekday

Critères de performance

- Le modèle choisi est un modèle de régression. Les critères pour évaluer la qualité d'un modèle de régression sont nombreux :
  - l'erreur moyenne absolue (MAE)
  - la racine carrée de la moyenne du carré des erreurs (RMSE)
  - le coefficient de détermination (R^2)
  - le critère d'information d'Akaike (AIC) ou bayésien (BIC).

- L'AIC ou le BIC permettent de pénaliser un modèle selon le nombre de paramètres entraînés. Cependant une validation croisée permet de comparer les scores directement sans avoir besoin de pénaliser pour le nombre de paramètres.
- **J'ai préféré utiliser le R^2** au MAE ou au RMSE, car ce critère donne des valeurs de référence connues : 1 quand le modèle donne des prédictions parfaites et 0 quand le modèle est aussi bon qu'un modèle naïf qui prédirait toujours la moyenne de la variable à prédire. La valeur du R^2 permet donc de repérer facilement les erreurs de code ou les problèmes d'entraînement.

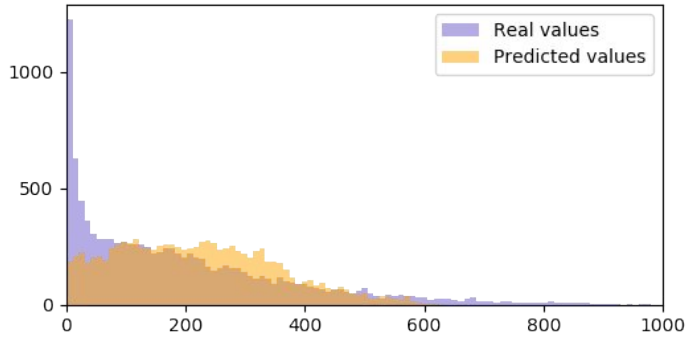
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

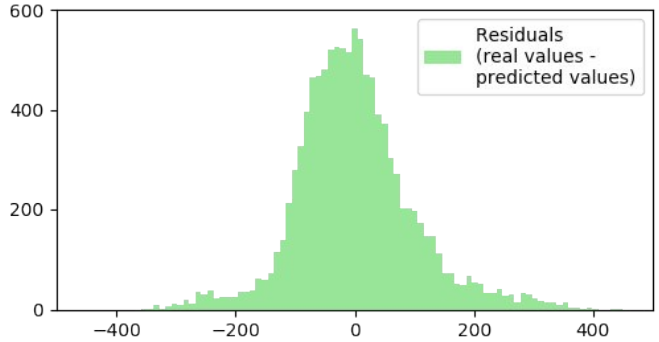
Performances du modèle

- Une moyenne du score a été réalisé sur les 5 scores issus de la validation croisée pour le modèle linéaire avec et sans les features additionnelles. On voit que **les features améliorent grandement** la prédiction du modèle.

	Modèle sans features additionnelles	Modèle avec features additionnelles
R^2	0,28	0,69

- Pour le modèle avec features additionnelles, on voit que **les valeurs prédites et réelles se superposent assez bien** (même si le modèle prédit mal les valeurs faibles). On voit également que les **résidus** du modèle ont bien l'air de suivre une **distribution gaussienne**.

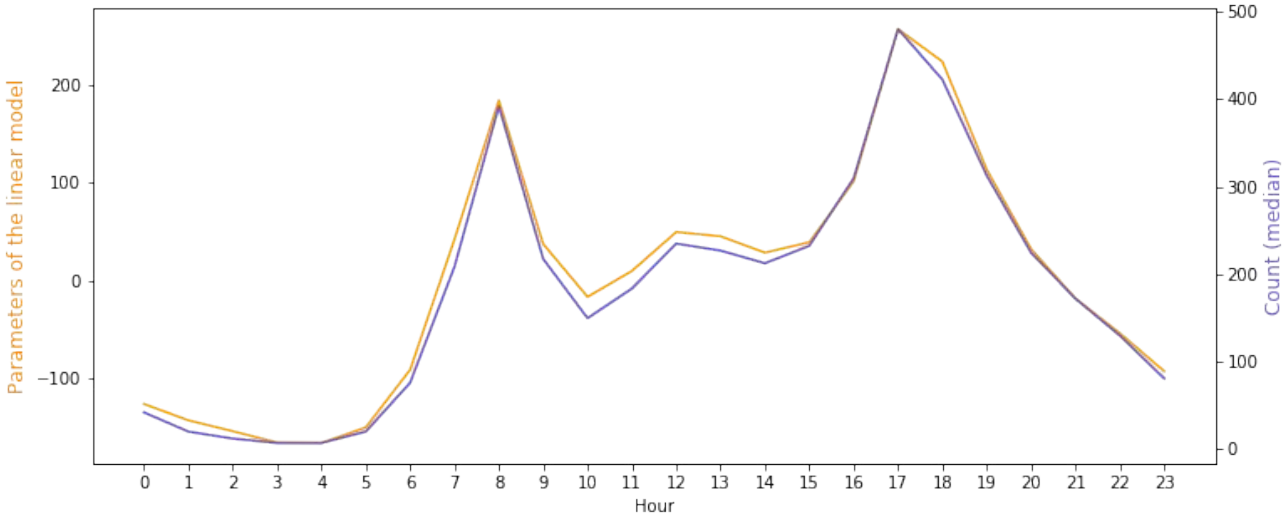




Paramètres du modèle

- On a déjà vu que la demande médiane en vélo dépend fortement de l'heure (en violet). J'ai donc voulu regardé les paramètres du modèle qui correspondent aux one-hot encodeurs de l'heure.

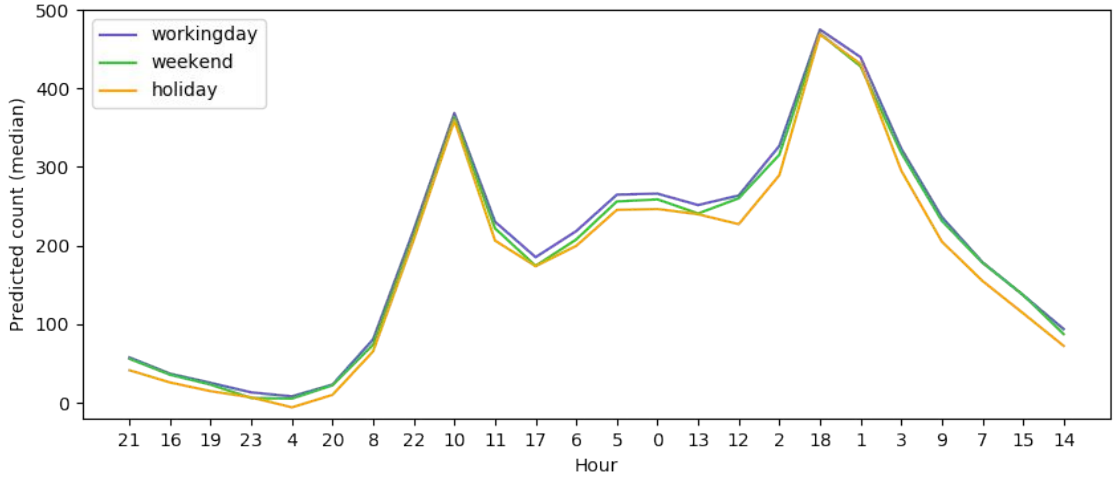
- On voit que les **paramètres** (en orange) prennent des valeurs qui **reproduisent le pattern des heures sur la demande réelle**, afin de prédire au mieux la demande :



Limites du modèle

- Nous avions vu dans la slide 10 qu'il y a une interaction entre le jour (travaillé, férié ou week-end) et l'heure sur la demande en vélos.

- Cependant une limite du modèle linéaire est qu'il peut seulement modéliser les effets linéaires entre les prédicteurs et la variable à expliquer. **Ce modèle ne peut donc pas**, entre autres, **reproduire l'effet d'interaction** qui existe dans les données :



Partie III :
Perspectives

Perspectives

- On a vu que le modèle développé n'a pas pu reproduire l'interaction observée entre l'heure et le jour sur les valeurs réelles.
  - Une perspective d'amélioration du modèle linéaire consisterait à réaliser une **régression par morceaux**. Il s'agit d'entraîner trois modèles différents selon si on est un jour travaillé, de week-end ou férié, et d'utiliser le modèle adéquat lors de la prédiction.
  - Il est également possible d'entraîner un random forest, un XGBoost ou un perceptron multicouches, des **modèles non linéaires**, qui pourraient arriver à reproduire l'effet de l'interaction

- Pour la suite du projet, il serait également pertinent de calculer les performances de **modèles adaptés aux séries temporelles**, tels que :
  - le modèle de Holt-Winters, qui prend en compte la tendance et la saisonnalité,
  - ARIMAX, qui peut également intégrer des prédicteurs externes,
  - un LSTM, pour lequel un certain travail de préparation des données est nécessaire.

Merci de votre attention

Héloïse THERO