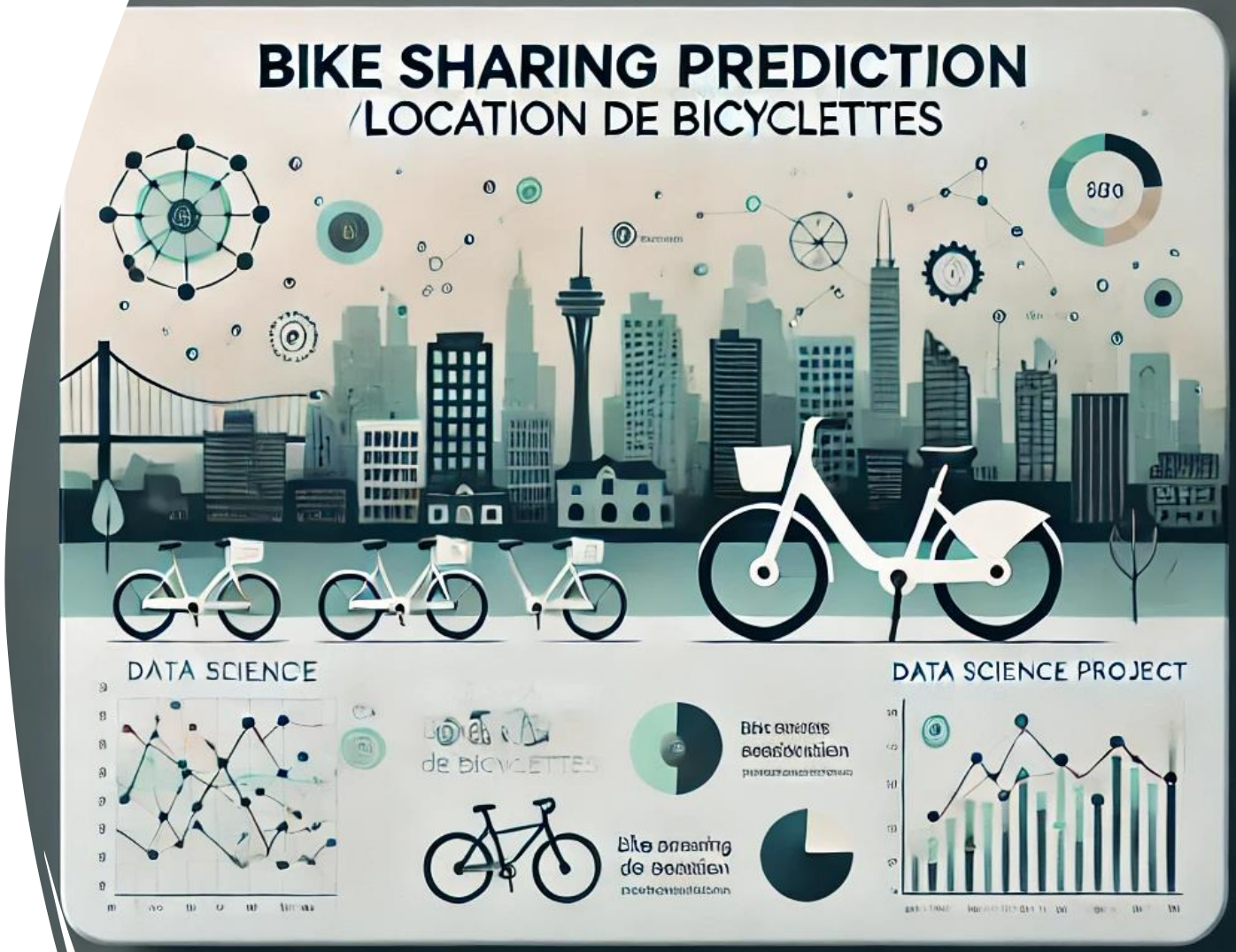



- Présentée par : OUANDJLI ASSIA
- Le 06/11/2024

# Plan de Travail

- **Résumé Exécutif**
- **Introduction**
- **Méthodologie**
- **Collection de donnée**
- **Modèles d’évaluation**
- **Prédiction et analyse**
- **Conclusion**

# Résumé Exécutif

- **Résumé des Méthodologies**
- Les données ont été collectées à partir d'un fichier CSV qui porte deux fichiers « Day & Hour .CSV ».
- location de bicyclettes dans la ville de Washington, D.C.
  sur la période 2011-2012.
- Des visualisations de données réalisée à partir des Modèles de machine Learning: Régression Linéaire(OLS), Lasso, Arbre de Décision , Forêt Aléatoire, Réseau neuronal séquentiel.
- **Résumé des Résultats**
- **Modèle optimal:** Forêt Aléatoire (Random Forest);
- **Test NMSE**: 0.15 / **$R^2$**: 0.85 (haute précision, généralisation solide)
- Avantage: excellent équilibre entre précision et généralisation

Random Forest explique 85 % de la variance des données de test, ce qui est le score le plus élevé parmi les modèles testés.

# Introduction

Dans ce projet, nous allons analyser les données de location de vélos, pour prédire la demande horaires et quotidienne d'utilisation des bicyclettes. Il est essentiel de savoir si nous pouvons anticiper la demande de manière précise, car une mauvaise allocation des vélos pourrait entrainer des coûts important pour les opérations de partage.

Les facteurs de jeu de données sont (season, yr, mnth, holiday, weekday, workingday, weathersit, temp, atemp, hum, windspeed, casual, registered, cnt)

**Problématiques à résoudre:**

- Quels facteurs influencent la demande de vélos?
- Peut-on prédire avec précision la demande de vélos?
- Quelle est la précision des prévisions concernant l'utilisation des vélos?

# Section 1: Methodologie

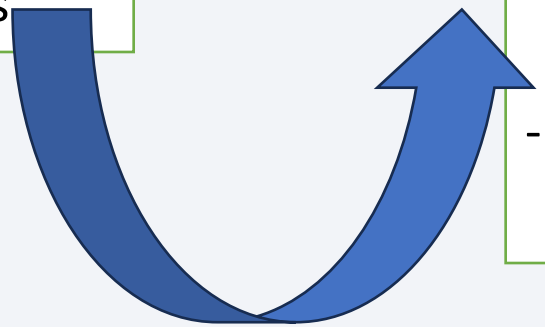
# Méthodologie

- **Jeu de données** : fichier CSV, utilisation des vélos contenant des données horaires et quotidiennes sur la location de vélos, incluant des variables telles que la météo, les jours fériés, et les saisons.
- **Data Cleaning:** Identifier et traiter les valeurs manquantes dans la dataset, conversion des dates/heures pour l'analyse temporelle, gestion des données aberrantes (outliers) , suppression des doublons.
- **Feature Engineering** : extraction de caractéristiques, variables liées à la météo, jours fériés, saisons, transformation des variables catégorielles en dummy variables (variable fictive), agrégation.
- **Modélisation et sélection des variables :**
- méthodes Backward et Forward.
- Fitting de modèles; (Régression OLS, LASSO, arbres de décision, forêt aléatoire, ARIMA, LSTM, GRU).
- Évaluation des performances (In sample & Out of sample)
- Métriques d'évaluation (MSE, Accuracy)
- Analyse prédictive: modèles de classification, prévision de la demande de vélos.
- **- Visualisation des données** : PCA, scatter plot, histogrammes

# Description des données

**1- Décompression du fichier:**

- Le fichier compressé a été extrait sur ce site; https://www.kaggle.com/datasets/lakshmi25npathi/bike-sharing-dataset pour récupérer les données au format CSV
- Les données comprenaient des enregistrements tels que l'utilisation des vélos, les conditions météorologiques, et les jours fériés



**2-Conversion en Data Frame à l'aide d'un parseur CSV;**

- Les données CSV ont été télécharger avec la bibliothèque Pandas
- Les données nettoyé et préparé pour l'analyse

- **Gestion de données**

1- Valeurs manquantes: aucune (data set complet)

2- j'ai effectué la conversion de la date et heures pour l'adapter au format naturel de l'analyse temporelle.

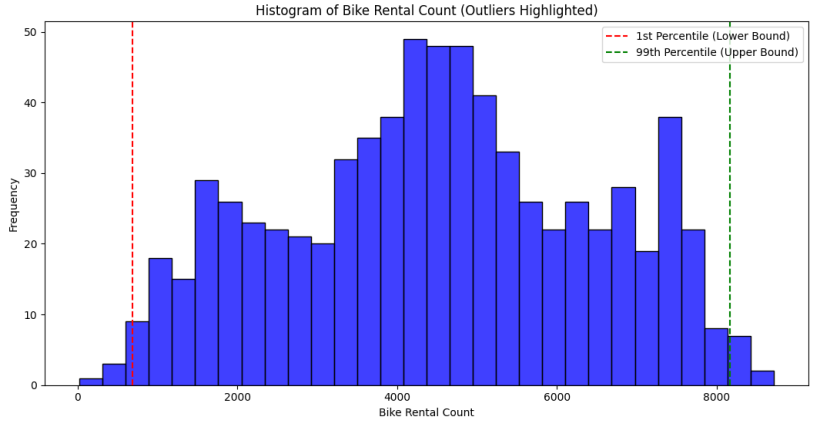
```
# Convert 'dteday' to datetime format
day_df['dteday'] = pd.to_datetime(day_df['dteday'])
hour_df['dteday'] = pd.to_datetime(hour_df['dteday'])

# Convert 'hr' to 12-hour format with AM/PM notation for hour_df
if 'hr' in hour_df.columns:
    hour_df['hour_12'] = hour_df['hr'] % 12
    hour_df['hour_12'] = hour_df['hour_12'].replace(0, 12)
    hour_df['period'] = np.where(hour_df['hr'] < 12, 'AM', 'PM')
```

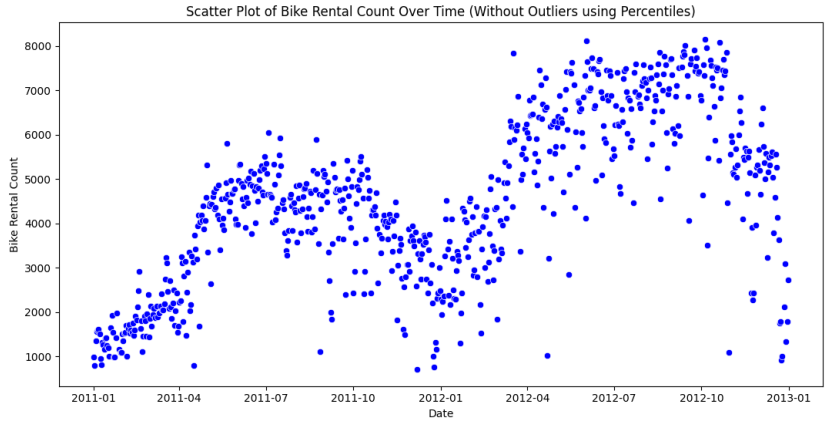
```
# Handle missing values in day_df
missing_values_day = day_df.isnull().sum()
missing_values_day = missing_values_day[missing_values_day > 0]
print("Missing values in day_df:")
print(missing_values_day)


# Handle missing values in hour_df
missing_values_hour = hour_df.isnull().sum()
missing_values_hour = missing_values_hour[missing_values_hour > 0]
print("\nMissing values in hour_df:")
print(missing_values_hour)
```

```
Missing values in day_df:
Series([], dtype: int64)

Missing values in hour_df:
Series([], dtype: int64)
```

# Détection des Données abérrantes (outliers)

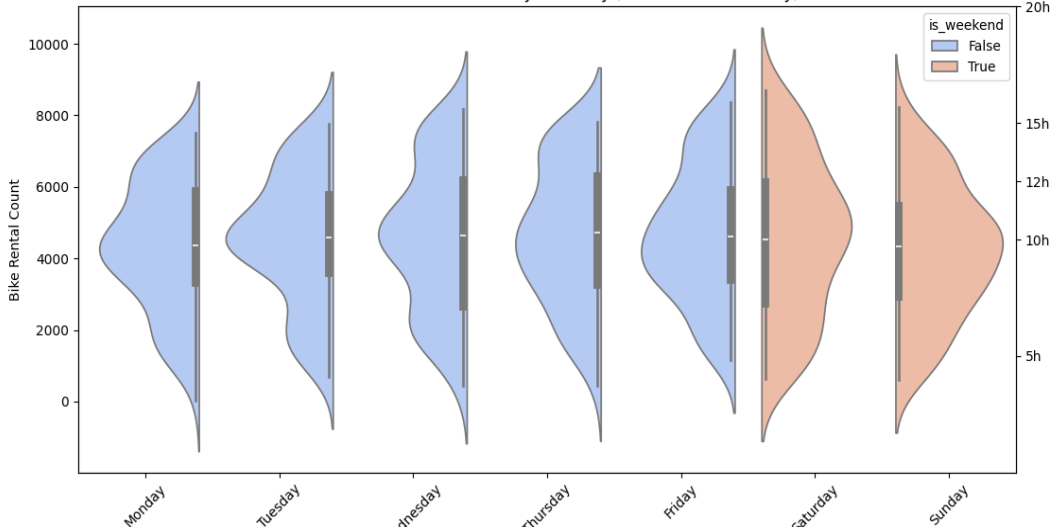
- Outliers: 16 détectés
- Méthodes: percentiles (1% -- 99%)
- Périodes affectées: ouragan, 2012-11-30
- Causes: Jours fériés, vacances
- Impact: Variation inhabituelles de la demande

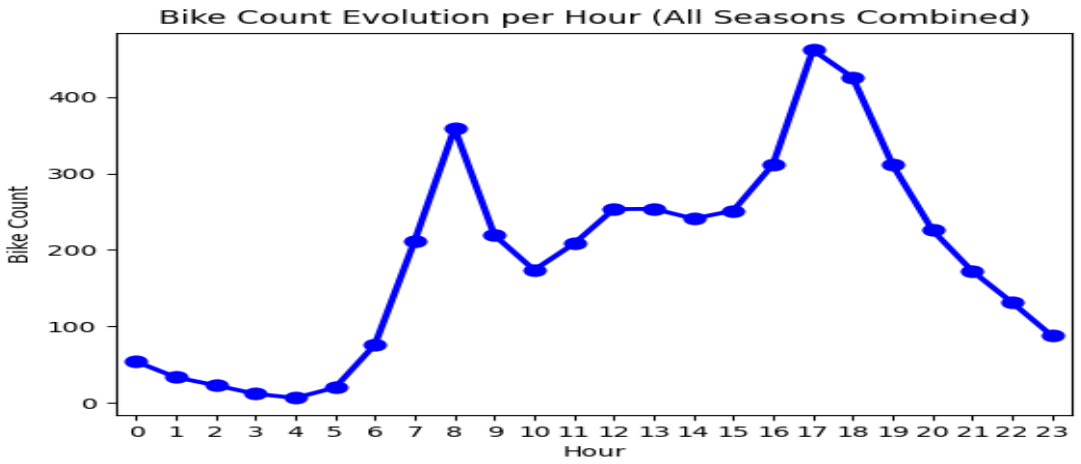


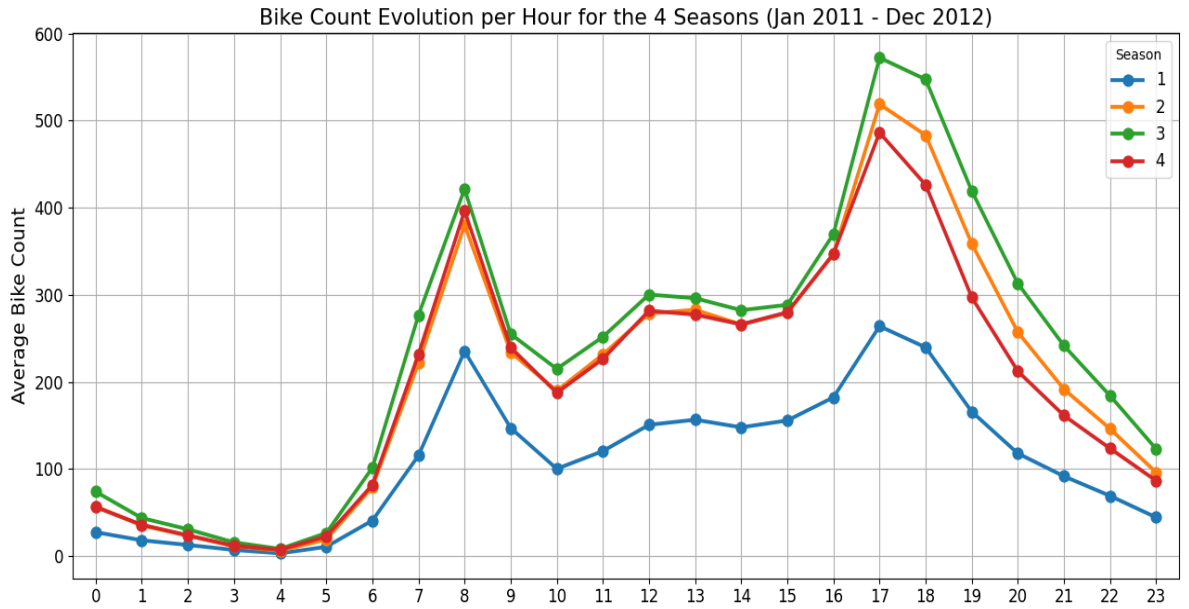



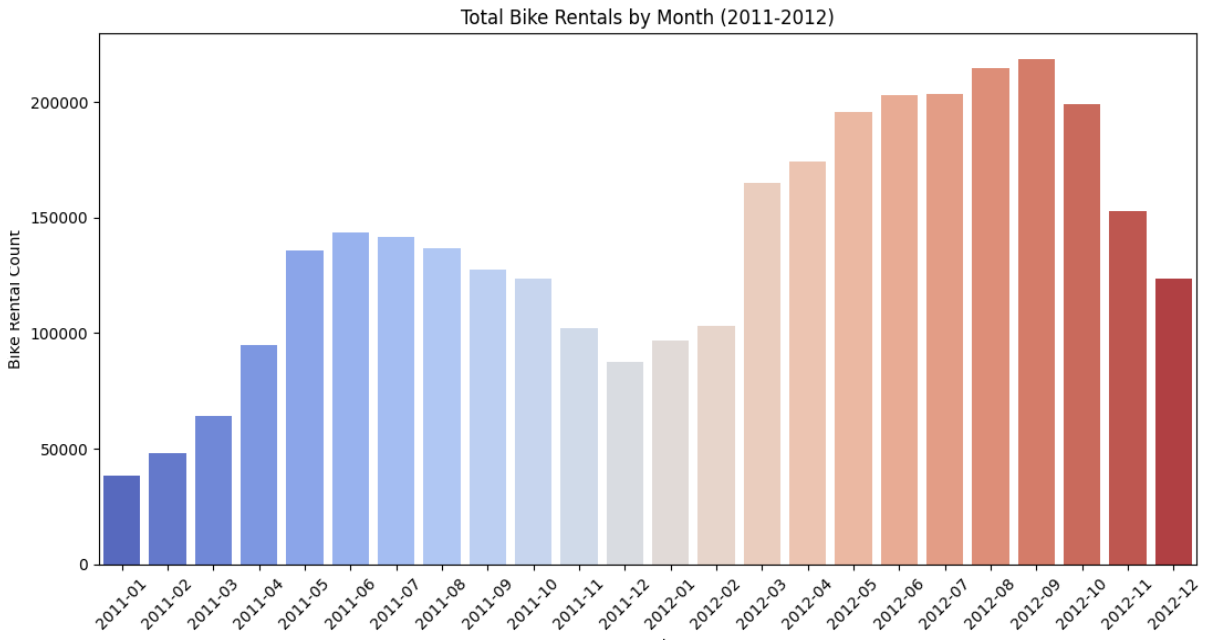

```python
# Détection des outliers avec les percentiles
lower_bound = day_df['cnt'].quantile(0.01)
upper_bound = day_df['cnt'].quantile(0.99)
# Filtrer les données pour enlever les outliers
df_without_outliers = day_df[(day_df['cnt'] >= lo
# Calculer le nombre d'outliers
number_of_outliers = len(day_df) - len(df_without_
print(f'Number of outliers filtered using percenti
```

```
Number of outliers filtered using percentiles: 16
```



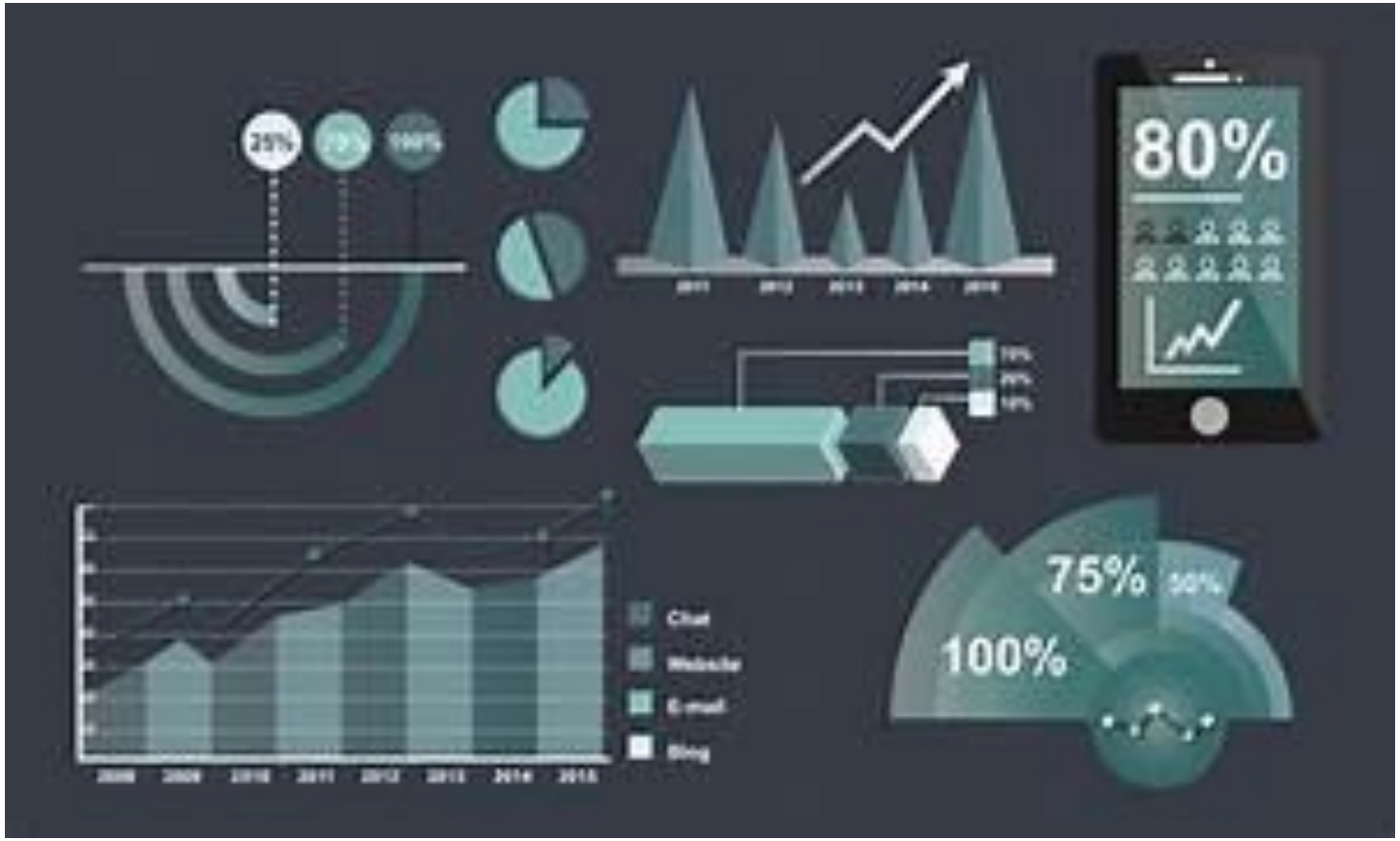

**Feature Engineering**
**Feature Aggregation** : par saison, mois, jour
**Mapping** : application d'un mapping sur les jours de la sem

**pics journaliers:** aller au travail, revenir du travail, week-en
Location été > location hiver

```python
# Convert 'dteday' to datetime format
day_df['dteday'] = pd.to_datetime(day_df['dteday'])
hour_df['dteday'] = pd.to_datetime(hour_df['dteday'])

# Convert 'hr' to 12-hour format with AM/PM notation for hour_df
if 'hr' in hour_df.columns:
    hour_df['hour_12'] = hour_df['hr'] % 12
    hour_df['hour_12'] = hour_df['hour_12'].replace(0, 12)
    hour_df['period'] = np.where(hour_df['hr'] < 12, 'AM', 'PM')
```

# La conversion des variables

Section 2: Modèles d’évaluation



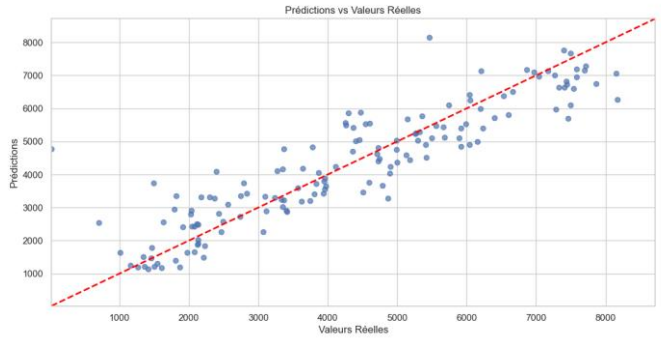
- La régression linéaire (OLS):
- Méthode Backward (Scatter plot)
Méthode Forward (PCA).
- NMSE: 0,19
- $R^2$: 0.81 (précision modérée)

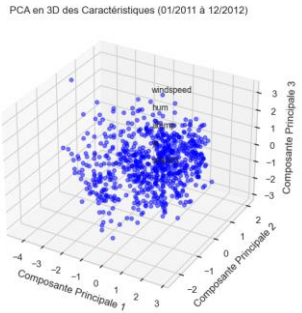
| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 1248.3209 | 272.690 | 4.578 | 0.000 | 712.724 | 1783.918 |
| season | 524.7225 | 65.596 | 7.999 | 0.000 | 395.885 | 653.560 |
| yr | 2023.9975 | 73.972 | 27.362 | 0.000 | 1878.708 | 2169.287 |
| mnth | -38.4447 | 20.537 | -1.872 | 0.062 | -78.783 | 1.893 |
| holiday | -391.5508 | 240.168 | -1.630 | 0.104 | -863.269 | 80.167 |
| weekday | 72.9370 | 18.310 | 3.984 | 0.000 | 36.975 | 108.900 |
| workingday | 160.8049 | 80.523 | 1.997 | 0.046 | 2.647 | 318.963 |
| weathersit | -632.8563 | 87.718 | -7.215 | 0.000 | -805.145 | -460.568 |
| temp | 2097.2478 | 1480.268 | 1.417 | 0.157 | -810.176 | 5004.672 |
| atemp | 3488.0422 | 1675.527 | 2.082 | 0.038 | 197.106 | 6778.979 |
| hum | -865.4394 | 354.490 | -2.441 | 0.015 | -1561.701 | -169.178 |
| windspeed | -2080.5404 | 519.038 | -4.008 | 0.000 | -3099.994 | -1061.087 |

| | | | |
|---|---|---|---|
| Omnibus: | 54.728 | Durbin-Watson: | 1.967 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 99.315 |
| Skew: | -0.595 | Prob(JB): | 2.72e-22 |
| Kurtosis: | 4.633 | Cond. No. | 519. |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

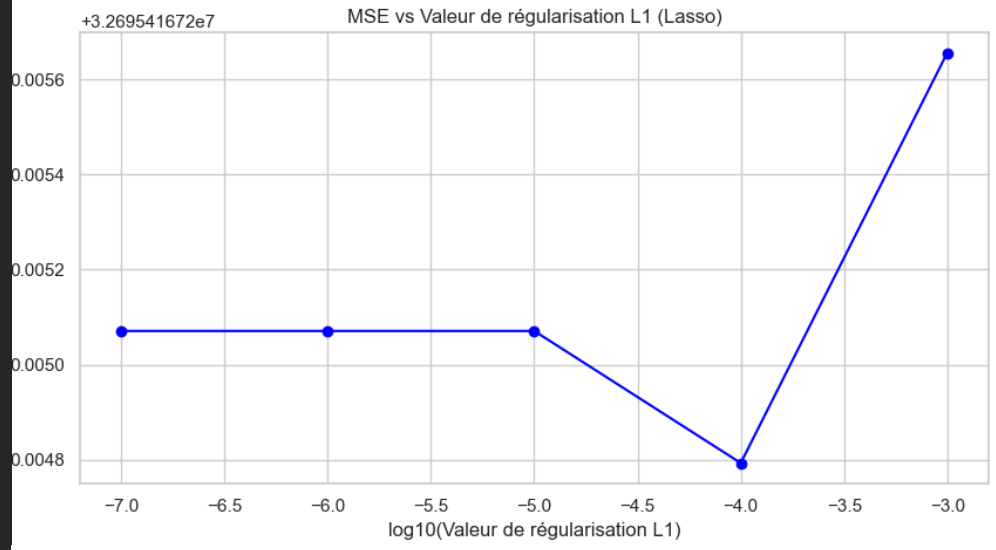
Analyse de régression (régression linéaire simple avec 'temp') :
Ordonnée à l'origine : 1272.16, Pente : 6575.39, $R^2$ : 0.39

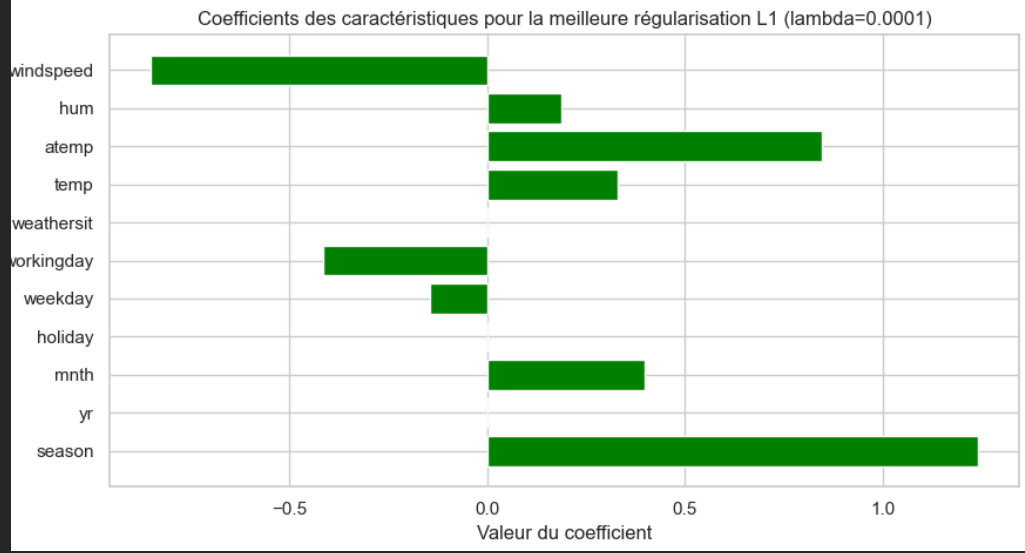



| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 1612.1144 | 265.384 | 6.075 | 0.000 | 1090.879 | 2133.350 |
| season | 425.5373 | 38.446 | 11.068 | 0.000 | 350.027 | 501.048 |
| yr | 1985.7183 | 78.247 | 25.378 | 0.000 | 1832.036 | 2139.400 |
| atemp | 6311.9026 | 259.169 | 24.354 | 0.000 | 5802.875 | 6820.931 |
| hum | -2597.4630 | 287.539 | -9.033 | 0.000 | -3162.212 | -2032.714 |
| windspeed | -2742.1300 | 528.375 | -5.190 | 0.000 | -3779.899 | -1704.361 |

| | | | |
|---|---|---|---|
| Omnibus: | 62.090 | Durbin-Watson: | 1.897 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 106.802 |
| Skew: | -0.682 | Prob(JB): | 6.43e-24 |
| Kurtosis: | 4.590 | Cond. No. | 44.3 |



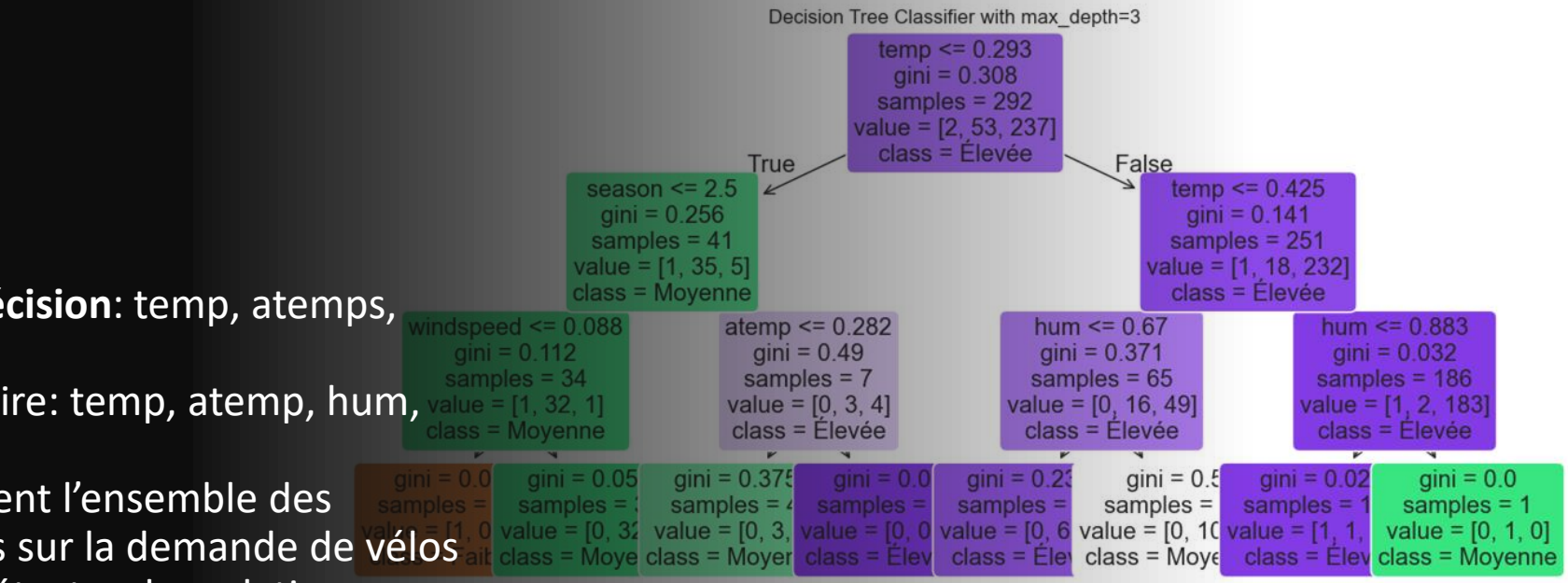




# LASSO Model(Pénalisation L1)

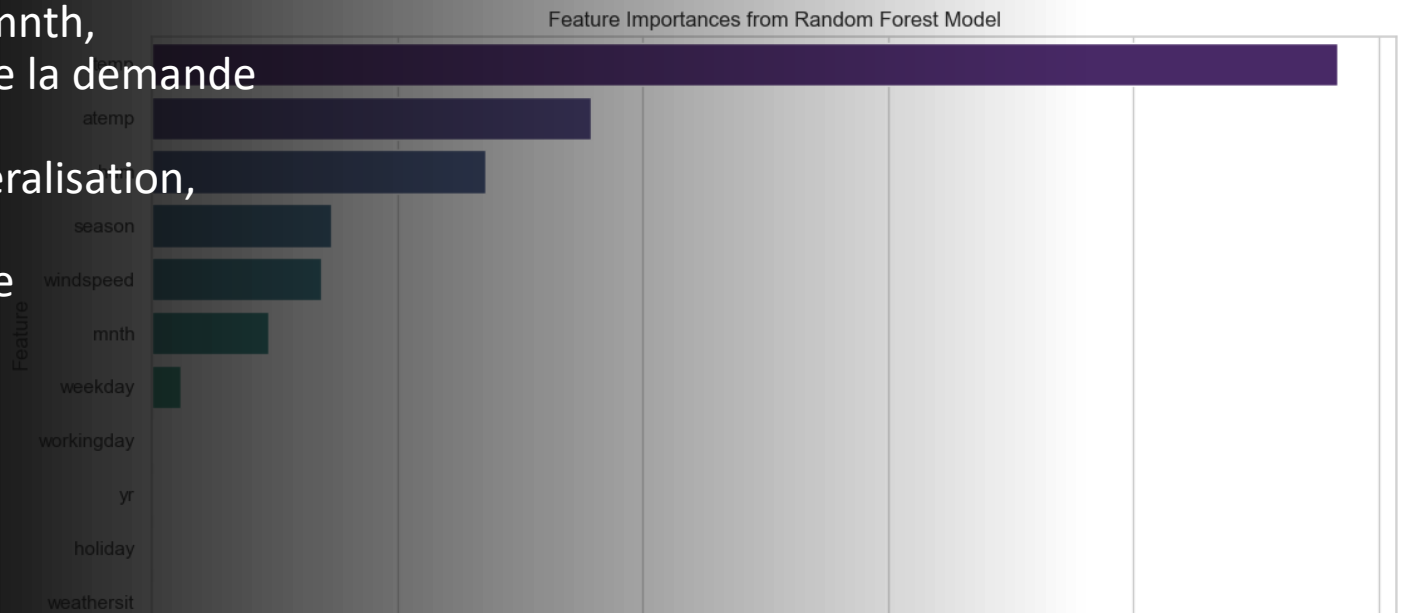
- **Variables Sélectionnées** : temp, atemp, hum, mnth, season
- **Métriques de Performance** :**MSE** : 8,15 (Indique une bonne précision)
- **$R^2$** : - 8,96 (indique que le modèle est inadapté aux données)
- **Pénalisation** : Diminue l'impact des variables moins importantes
- **Objectif** : Réduire le surapprentissage
- **Approche** : Conserver un large éventail de variables pour une modélisation complète
- **Utilité** : Capturer l'influence des conditions météorologiques (temp, atemp et hum), et des aspects temporels (Mnth, Season) sur la demande des vélos

L’Arbre de decision et Forêt Aléatoire

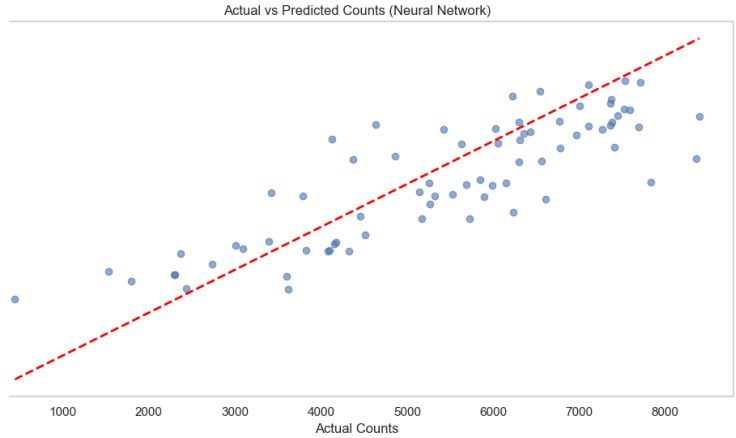
**Variables sélectionnées Arbre de décision**: temp, atemps, season hum, windspeed
variables sélectionnées Forêt Aléatoire: temp, atemp, hum, season, windspeed, mnth, weekday
**Interprétation: Ces** variables capturent l’ensemble des influences climatique et temporelles sur la demande de vélos
**Avantages des arbres:** capable de détecter des relations non linéaires / Robustesse face aux données bruitées.

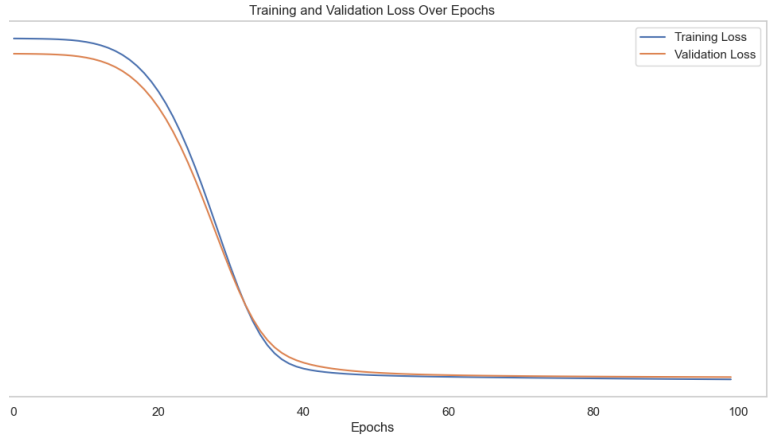


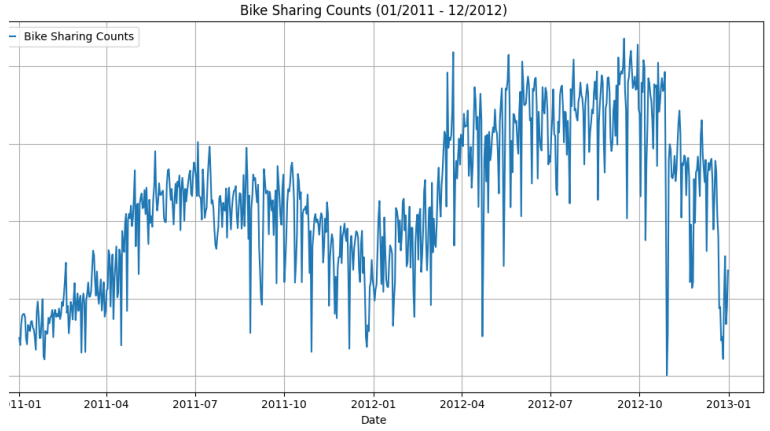

**Insight:** les conditions météorologique (temp, atemp, hum, windspeed) et les facteurs temporelles (season, mnth, weekday) sont déterminants pour la prédiction de la demande
**Résultats:**
Forêt Aléatoire (Précision élevée et Meilleur généralisation, MSE Test; 0,15/ $R^2$; 0,85)
Arbre de décision; Bonne précision mais risque de surapprentissage MSE Test: 0,51/ $R^2$: 0,49



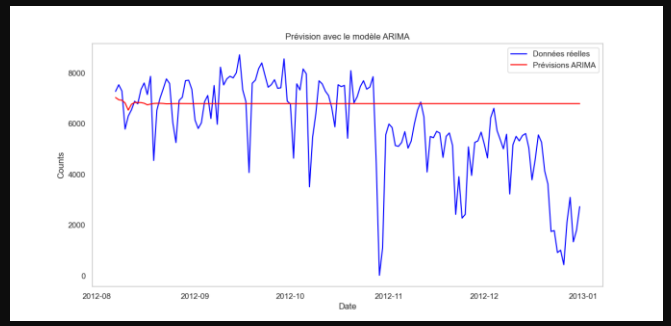
Réseau Neuronal Séquentiel:
Performance:
MSE: 0,30
$R^2$: 0,67
Interprétation: precision limitée par rapport au Random Forest, expliquant 68% de la variance des données de test.
Optimisation; Diminution progressive du MSE avec l'augmentation du nombre d'epoch
Courbe d'erreur: tendance décroissante montrant une amelioration continue avec l'entrainement, mais convergence à un niveau de precision inférieur au Random Forest

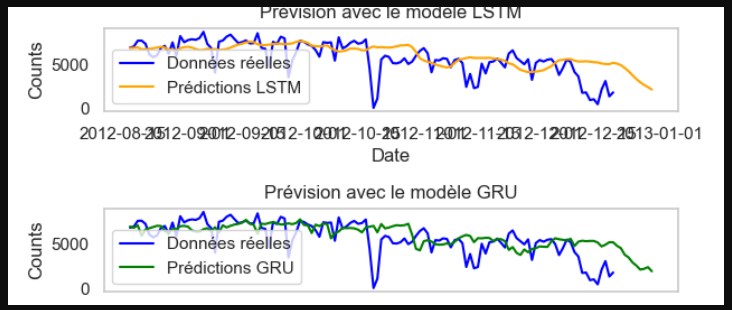


# Arima, LSTM, GRU Models:




ARIMA :
MSE : 0,30
$R^2$ : -0.23 (Score négatif)
Performance : Faible adaptation aux données, le modèle ne capte pas suffisamment la complexité des tendances.

LSTM :
MSE : 0,30
$R^2$ : -741736344,77 (Erreur extrême)
Fiabilité : Prédictions incohérentes, erreurs très élevées, modèle inapproprié pour cette tâche.

GRU :
MSE : 0,30
$R^2$ : -748054013,95 (Score négatif extrême)
Efficacité : Échec similaire à LSTM, inadapté pour capture la structure des données de demande de vélos.

# Analyse:

1.Facteurs influençant la demande de vélos :

Conditions météorologiques : température, ressenti de température, vent
Saisons : hiver, été, etc.
Jours : ouvrés, week-ends, jours fériés, vacances
Variables clés : saison, température, usagers occasionnels (casual)
Impact notable : saison avec coefficient de 331.56 (régression)

2. Précision des prédictions :

Modèle le plus performant : Random Forest est souvent performant pour des données ayant des relations non linéaires, ce qui pourrait bien être le cas pour la demande de vélos.
Random Forest Training MSE : 0,03, Ce MSE est très bas, indiquant que le modèle s'ajuste bien aux données d'entraînement, mais cela peut aussi indiquer un risque de surapprentissage (overfitting).
Random Forest Test MSE: 0,15, Un test MSE plus élevé que le MSE d'entraînement est normal et peut indiquer une bonne généralisation.
Random Forest $R^2$ : 0.85, signifie que le modèle explique environ 85 % de la variance de la demande de vélos, ce qui est un excellent score et montre une forte capacité du modèle à capturer les tendances.

# Conclusion:

Pour conclure on constate que, l'analyse des modèles de prédiction de la demande de vélos a mis en évidence plusieurs points clés. Les facteurs climatiques et saisonniers jouent un rôle déterminant dans la variation de la demande, nécessitant des modèles capables de capturer ces aspects complexes.

Parmi les différents modèles évalués, la Forêt Aléatoire (Random Forest) s'est révélée la plus efficace. Elle offre une précision de prédiction supérieure aux autres modèles, bien qu'il reste une marge d'amélioration.

Objectif futur : Optimiser davantage le modèle pour affiner la précision des prévisions, en ajustant ses hyperparamètres et en intégrant éventuellement d'autres variables explicatives pour capturer au mieux la dynamique de la demande de vélos.