

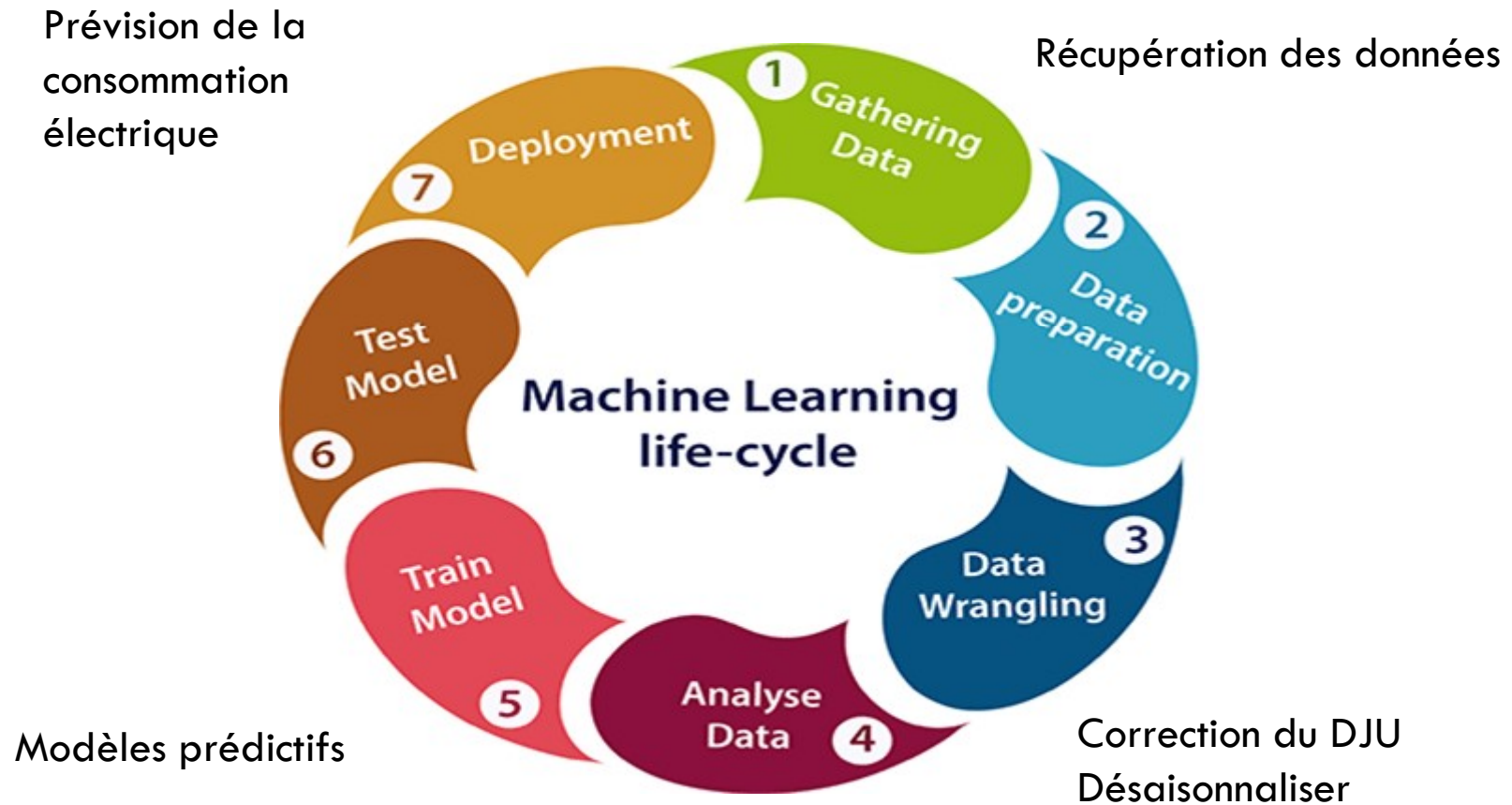
Prédire la consommation en électricité

Année 2021-2022



Sommaire

2



Problématique

3



La plupart des énergies renouvelables est intermittente.



La demande en électricité des utilisateurs varie au cours du temps, et dépend de paramètres comme la météo.



Optimiser les ressources pour mieux les distribuer

EMPLOYÉ CHEZ ENERCOOP



Récupération des données

4



Récupération des données

5

- Consommation électrique
- De janvier 2012 à novembre 2021

	Mois	Conso_totale
0	2012-01-01	51086
1	2012-02-01	54476
2	2012-03-01	43156
3	2012-04-01	40176
4	2012-05-01	35257

```
consommation_france=df.loc[df['Territoire']=='France',['Mois','Consommation totale']]
```

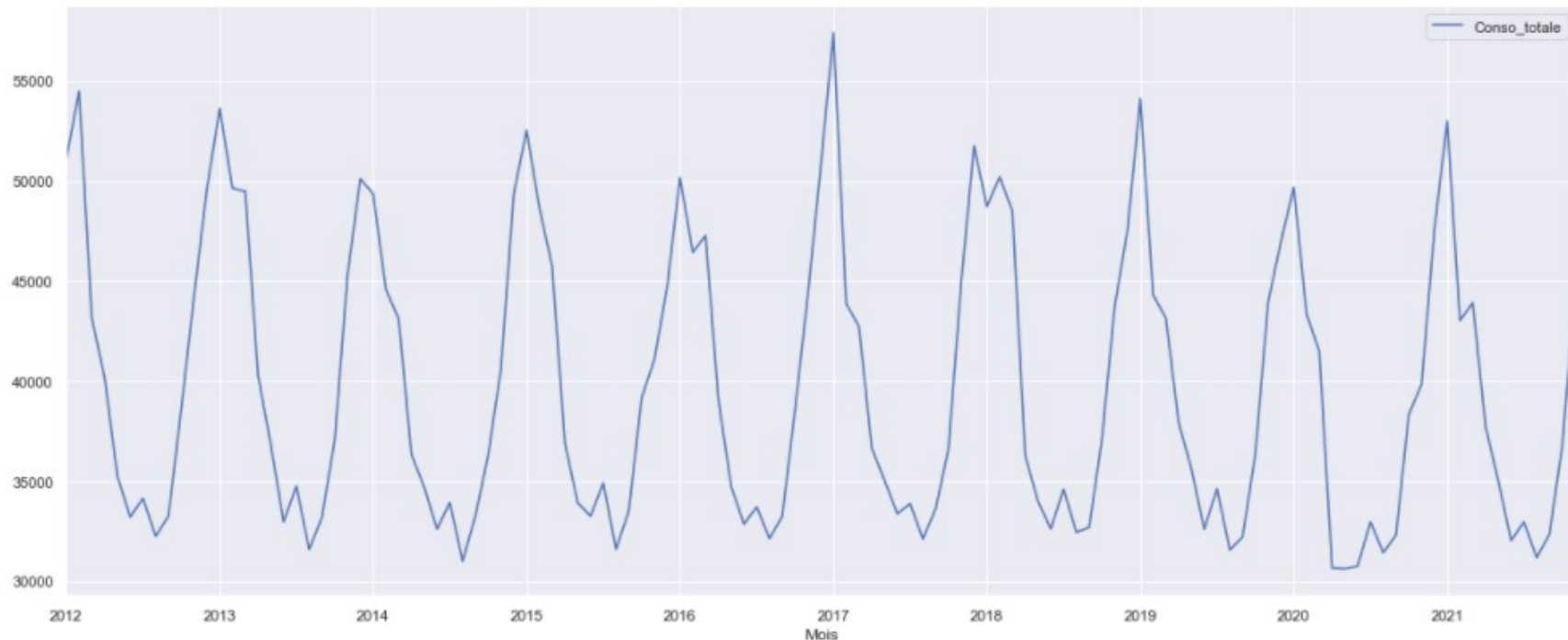


Figure 1 : Consommation d'électricité en France au cours du temps

Récupération des données

6

- DJU
- Les données récupérées pour le calcul du DJU proviennent de CEGIBAT.
- Une moyenne nationale a été réalisée en récupérant les données des différents territoires.

■ 2. Sélectionnez la méthode de calcul ([en savoir plus sur les méthodes de calculs](#))

☐ Météo ☒ Professionnels de l'énergie

■ 3. Sélectionnez le type d'usage

☒ Chauffage ☐ Climatisation

■ 4. Sélectionnez la température de référence



Degré jour unifié



Le degré jour unifié est la différence entre la température extérieure et une température de référence qui permet de réaliser des estimations de consommations d'énergie thermique pour maintenir un bâtiment confortable en proportion de la rigueur de l'hiver ou de la chaleur de l'été. [Wikipédia](#)

Récupération des données

7

□ Dataframe initial

	Unnamed: 0	JAN	FÉV	MAR	AVR	MAI	JUN	JUI	AOÛ	SEP	OCT	NOV	DÉC	Total
0	2021	0	0.5	5.7	4.9	13.9	54.1	0.0	0.0	0.0	0.0	0.0	0	79.2
1	2020	0	0.0	0.6	35.7	48.8	74.6	112.2	165.1	84.1	1.4	0.5	0	523.1
2	2019	0	0.5	1.2	14.5	12.0	111.3	162.3	125.7	41.2	7.2	0.0	0	476.1
3	2018	0	0.0	0.1	25.3	61.1	86.7	200.3	136.5	52.6	28.0	0.3	0	590.7
4	2017	0	0.0	2.8	4.5	60.0	123.9	120.8	94.5	20.3	12.7	0.0	0	439.5
5	2016	0	0.0	0.0	2.0	21.1	50.3	114.6	132.0	74.8	1.4	0.0	0	396.1

□ Transformation

```
meteo_newformat={'mois':[],'climatisation':[]}  
  
for Année in meteo.index.values:  
    for mois in meteo.columns:  
        meteo_newformat['mois'].append(f"{Année}-{mois}-01")  
        meteo_newformat['climatisation'].append(meteo.loc[Année,mois])  
  
meteo_newformat=pd.DataFrame(meteo_newformat)  
meteo_newformat['mois']=pd.to_datetime(meteo_newformat['mois'])
```

	mois	climatisation
0	2021-01-01	0.057143
1	2021-02-01	0.828571
2	2021-03-01	4.328571
3	2021-04-01	7.928571
4	2021-05-01	19.828571

□ Dataframe final

	Conso_totale	dju
Mois		
2012-01-01	51086	349.000000
2012-02-01	54476	444.428571
2012-03-01	43156	229.614286
2012-04-01	40176	209.028571
2012-05-01	35257	136.485714

Récupération des données

8

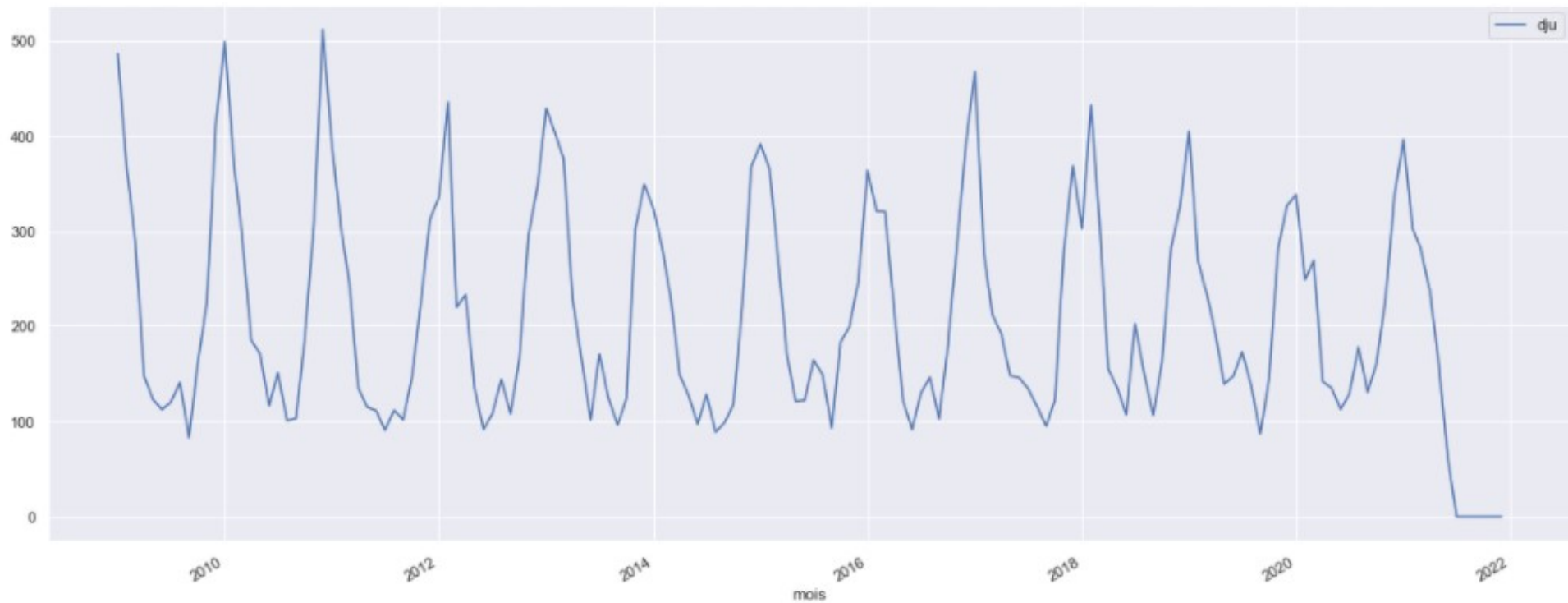


Figure 2 : Evolution du DJU au cours du temps

Comparaison

9

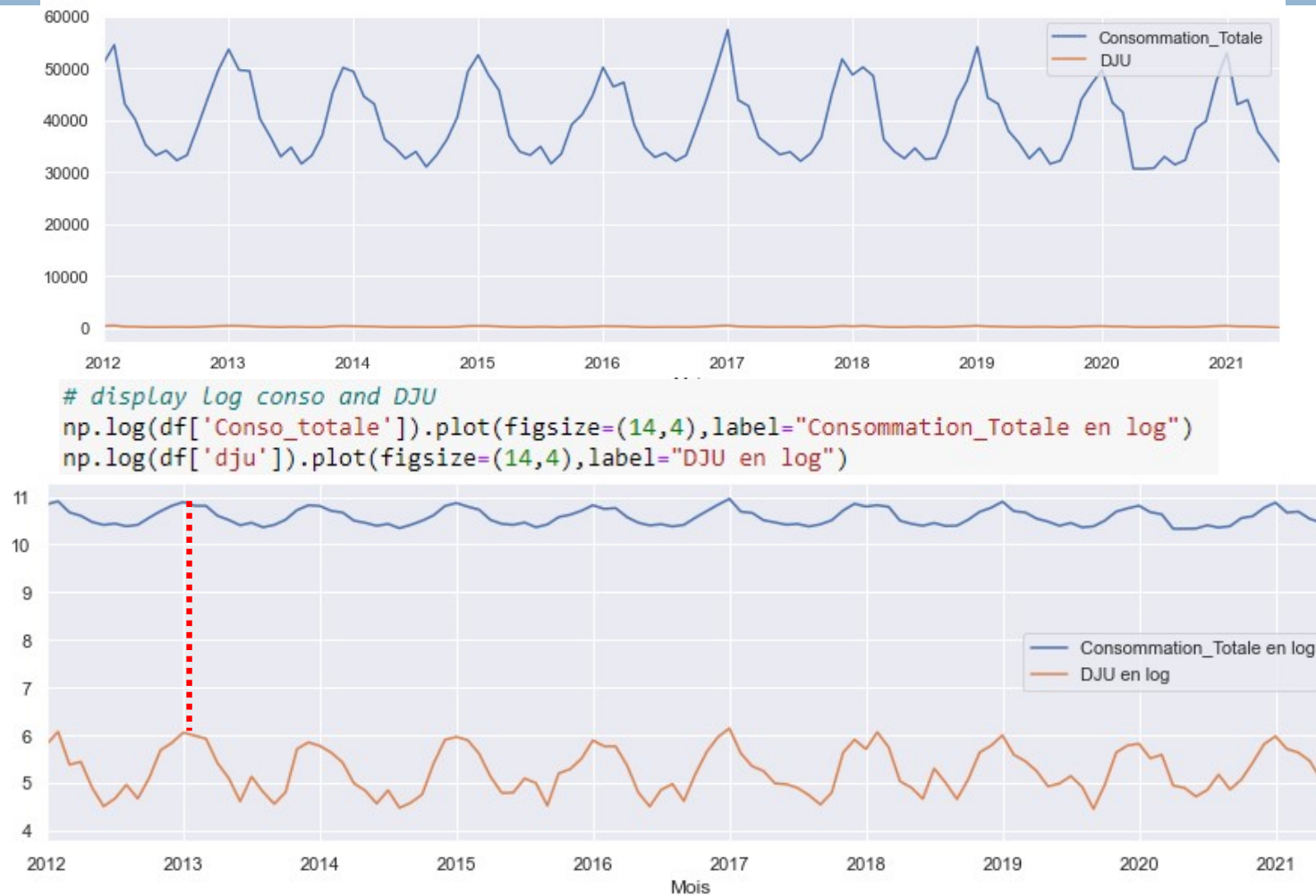
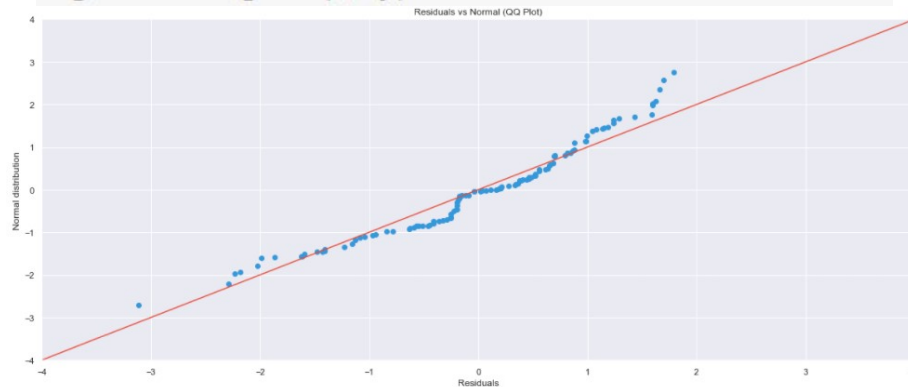


Figure 3 : Comparaison de la consommation électrique et du DJU sous forme normale ou logarithmique

Correction de la DJU

10

```
reg = linear_model.LinearRegression(fit_intercept=True)
reg.fit(x, y)
regression=reg.fit(x, y)
```



```
print(reg.coef_, reg.intercept_)
c = reg.coef_[0]
serie_corrige = df['Conso_totale'] - df['dju']*c
[66.74926459] 25452.86105647532
```

Nos résidus suivent une loi normale

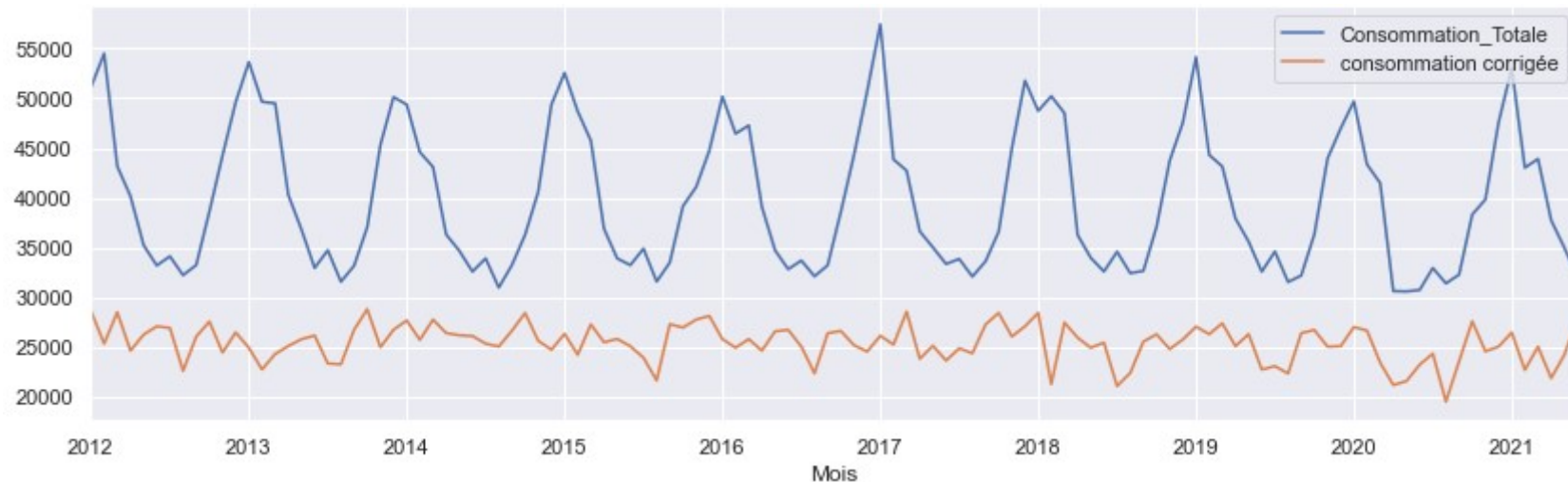
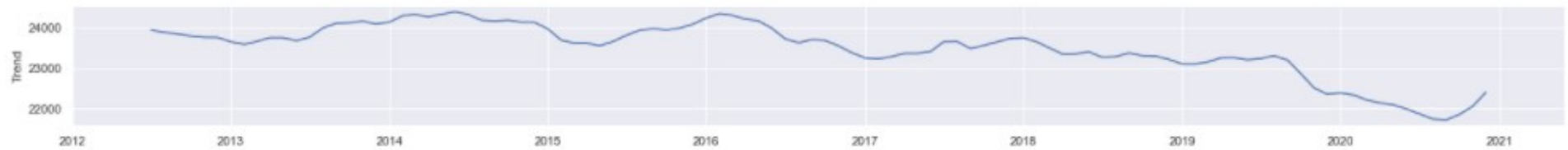


Figure 4 : Consommation électrique totale et sa correction en supprimant l'effet du DJU

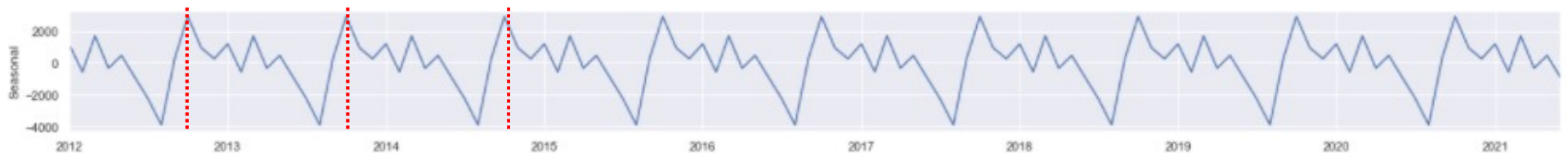
Moyenne Mobile : modèle additif



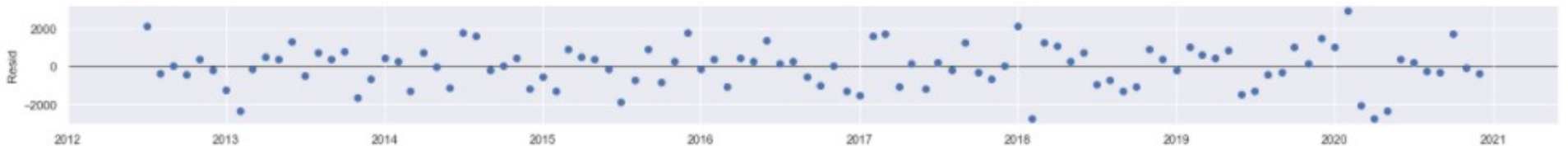
Série initiale, corrigée avec le DJU



Tendance : la série est en baisse constante irrégulière, il n'y a donc une tendance à la baisse



Saisonnalité : période de saisonnalité de 12 mois (période = S)



Résidu : qui est égal à la série corrigée moins l'addition de la tendance et de la saisonnalité

Figure 5 : Observation, Tendance, Effet saisonnier et Résidus du modèle additif de moyenne mobile

Moyenne mobile

12

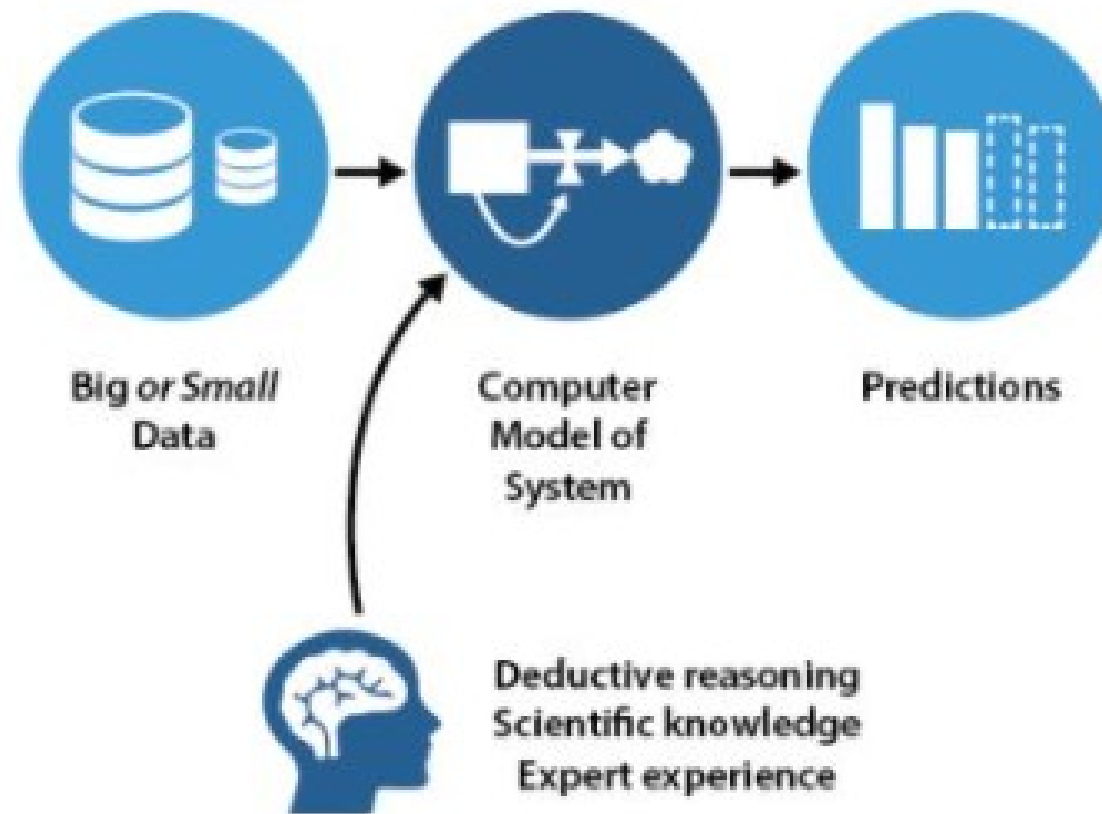
```
serie_corr_df['corrigee_desaison'] = serie_corrgee.values - decomp_x.seasonal.values  
plt.plot(serie_corr_df["corrigee"],'black', label="Consommation corrigée",alpha=0.25)  
plt.plot(serie_corr_df["corrigee_desaison"],'b', label="Consommation corrigée désaisonnalisée")  
  
plt.legend()  
plt.show()
```



Figure 6 : Evolution de la consommation corrigée et de la consommation corrigée désaisonnalisée

Modèle prédictif

13



Modèle Prédictif

14

Méthodologie pour les modèles



Modèle prédictif : Hotwinter

15

```
MAE=np.abs(x_a_prevoir['corrigee']-x_a_prevoir['prediction']).mean()
```

MAE

1077.7032



Figure 7 : Evolution de la consommation électrique désaisonnalisée ainsi que la valeur prédite par le modèle Hotwinter

Modèle prédictif : Hotwinter

16

```
MAE=np.abs(x_a_prevoir['corrigees']-x_a_prevoir['prediction']).mean()  
MAE  
1077.7032
```

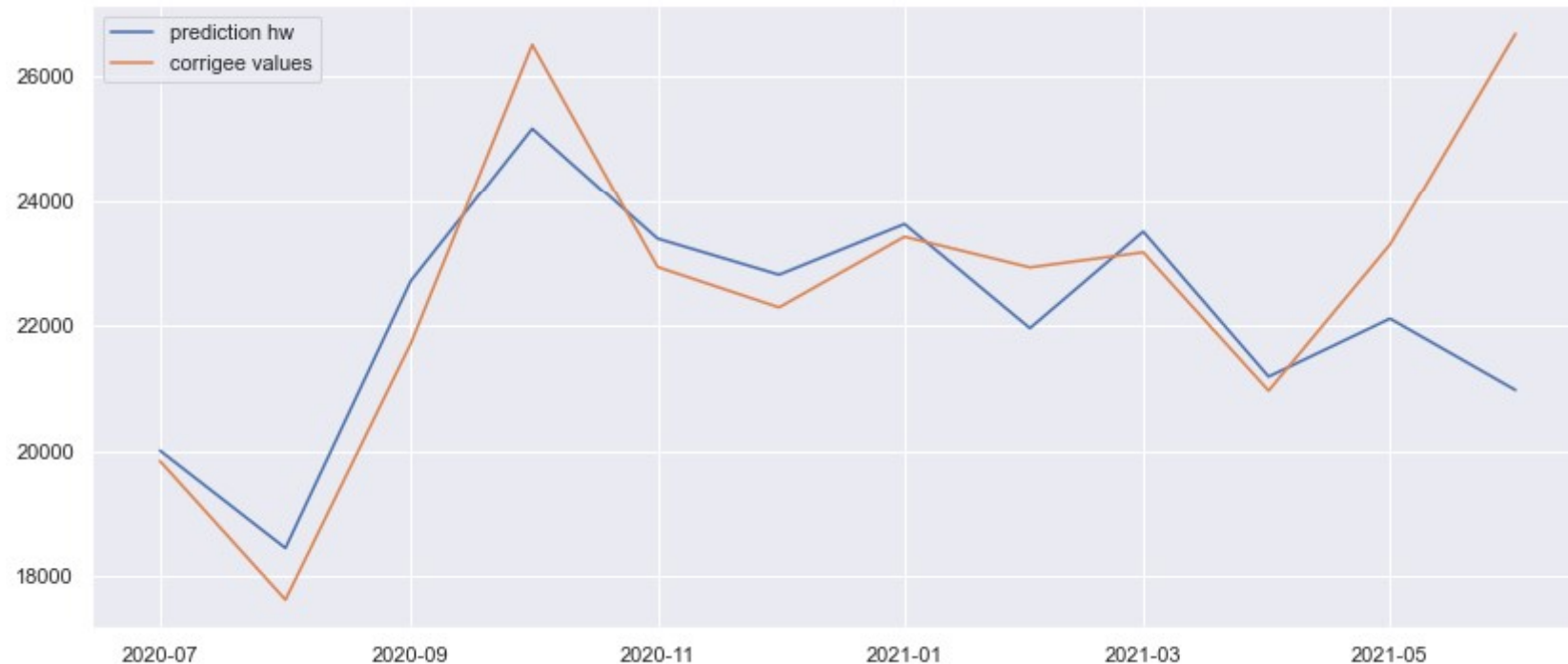


Figure 8 : Comparaison de la valeur réelle et de la valeur prédite par le modèle Hotwinter

Modèle prédictif : SARIMA

17

SARIMA

Sarima : $(p,d,q)(P,D,Q) [S]$

S

S=notre période

p

P

d

D

q

Q

Modèle prédictif : SARIMA 1

18

```
model1 = SARIMAX(np.asarray(x_tronc["corrige"]), order=(1,0,1), seasonal_order=(1,1,1,12))
results1 = model1.fit()
sarima_pred=results1.forecast(12)
```

SARIMAX Results

```
=====
Dep. Variable:          y      No. Observations:          102
Model:          SARIMAX(1, 0, 1)x(1, 1, 1, 12)      Log Likelihood          -800.276
Date:          Tue, 05 Apr 2022      AIC          1610.552
Time:          11:28:28      BIC          1623.051
Sample:          0      HQIC          1615.593
          - 102
Covariance Type:          opg
=====
```

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	-0.2866	2.653	-0.108	0.914	-5.487	4.914
ma.L1	0.3236	2.665	0.121	0.903	-4.901	5.548
ar.S.L12	-0.6024	0.336	-1.793	0.073	-1.261	0.056
ma.S.L12	0.5503	0.358	1.537	0.124	-0.152	1.252
sigma2	3.402e+06	5.87e+05	5.793	0.000	2.25e+06	4.55e+06

```
=====
Ljung-Box (Q):          41.33      Jarque-Bera (JB):          2.02
Prob(Q):          0.41      Prob(JB):          0.36
Heteroskedasticity (H):          2.35      Skew:          -0.36
Prob(H) (two-sided):          0.02      Kurtosis:          2.92
=====
```

Modèle prédictif : SARIMA 1

19

```
MAE_sarima_model2=np.abs(x_a_prevoir['corrige']-pred_tronc).mean()  
MAE_sarima_model2  
1617.3527
```



Figure 9 : Evolution de la consommation électrique désaisonnalisée ainsi que la valeur prédite par le modèle SARIMA 1

Modèle prédictif : SARIMA 1

20

```
MAE_sarima_model2=np.abs(x_a_prevoir['corrigees']-pred_tronc).mean()  
MAE_sarima_model2  
1617.3527
```



Figure 10 : Comparaison de la valeur réelle et de la valeur prédite par le modèle SARIMA 1

Modèle prédictif : SARIMA 2

21

```
pred_tronc = smodel.predict(12)

MAE_sarima_model_optim=np.abs(x_a_prevoir['corrige']-pred_tronc).mean()
MAE_sarima_model_optim
```

1615.3785746542214

```
# Seasonal - fit stepwise auto-ARIMA
smodel = pm.auto_arima(x_tronc["corrige"],
                      test='adf',
                      m=12,
                      seasonal=True,
                      d=0, D=1, trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)
```

Best model: ARIMA(2,0,2)(2,1,0)[12]

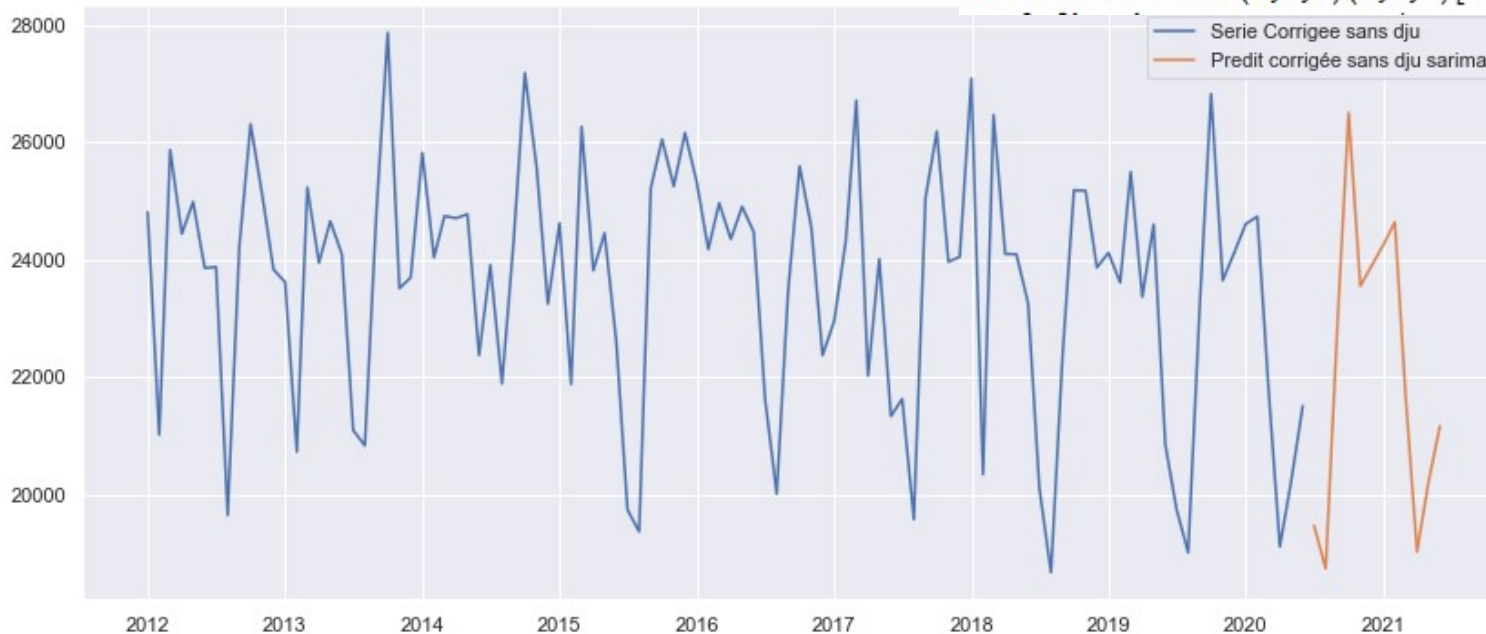


Figure 11 : Evolution de la consommation électrique désaisonnalisée ainsi que la valeur prédite par le modèle SARIMA 2

Modèle prédictif : SARIMA 2

22

```
pred_tronc = smodel.predict(12)

MAE_sarima_model_optim=np.abs(x_a_prevoir['corrige']-pred_tronc).mean()
MAE_sarima_model_optim
```

1615.3785746542214

```
# Seasonal - fit stepwise auto-ARIMA
smodel = pm.auto_arima(x_tronc["corrige"],
                      test='adf',
                      m=12,
                      seasonal=True,
                      d=0, D=1, trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)
```

Best model: ARIMA(2,0,2)(2,1,0)[12]



Figure 12 : Comparaison de la valeur réelle et de la valeur prédite par le modèle SARIMA 2

Comparaison des modèles prédictifs

23

```
MAE SARIMA1
1617.3527866265424
MAE SARIMA2
1615.3785746046808
MAE Hotwinter
1077.7032981758773    moyenne des écart Hotwinter
456.16259170154564
```

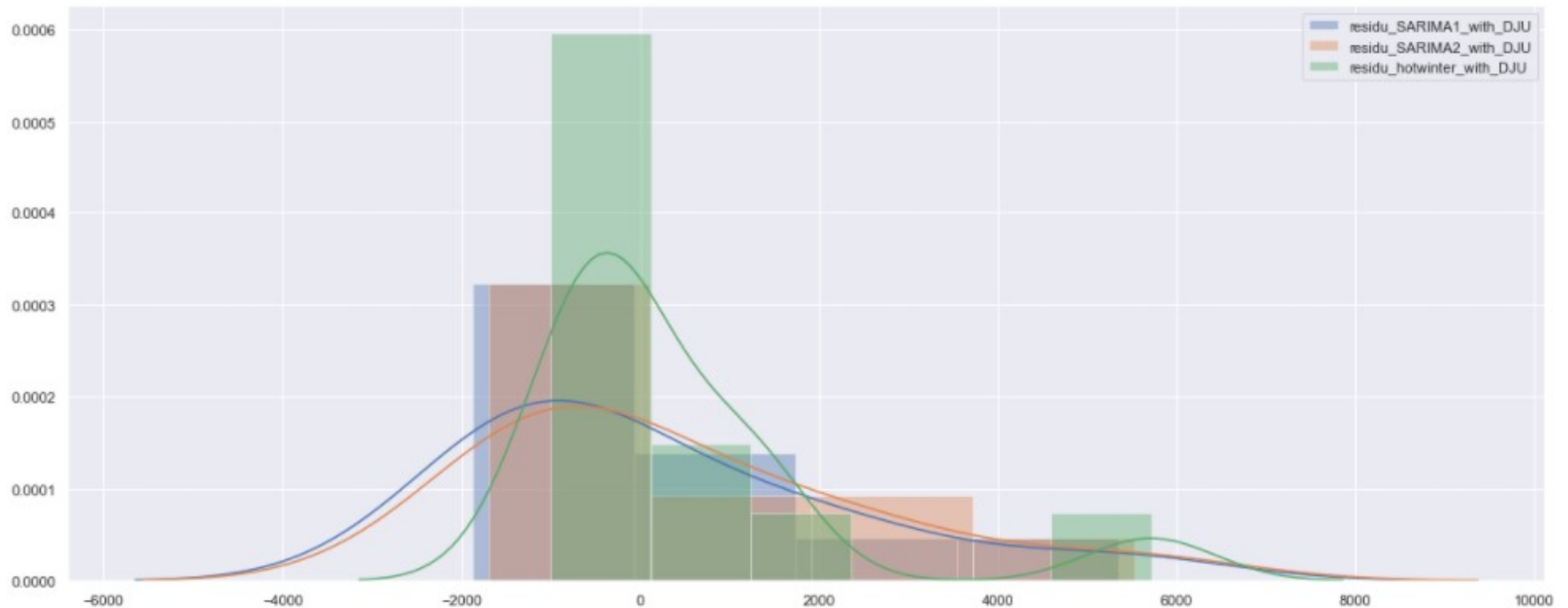


Figure 13 : Comparaison des résidus des différents modèles prédictifs

Modèle prédictif : Hotwinter

24

```
MAE=np.abs(x_a_prevoir['corrigees']-x_a_prevoir['prediction']).mean()  
MAE  
1077.7032
```



Figure 8 : Comparaison de la valeur réelle et de la valeur prédite par le modèle Hotwinter

Modèle prédictif pour l'année à venir

25



Modèle prédictif Hotwinter pour l'année à venir

26

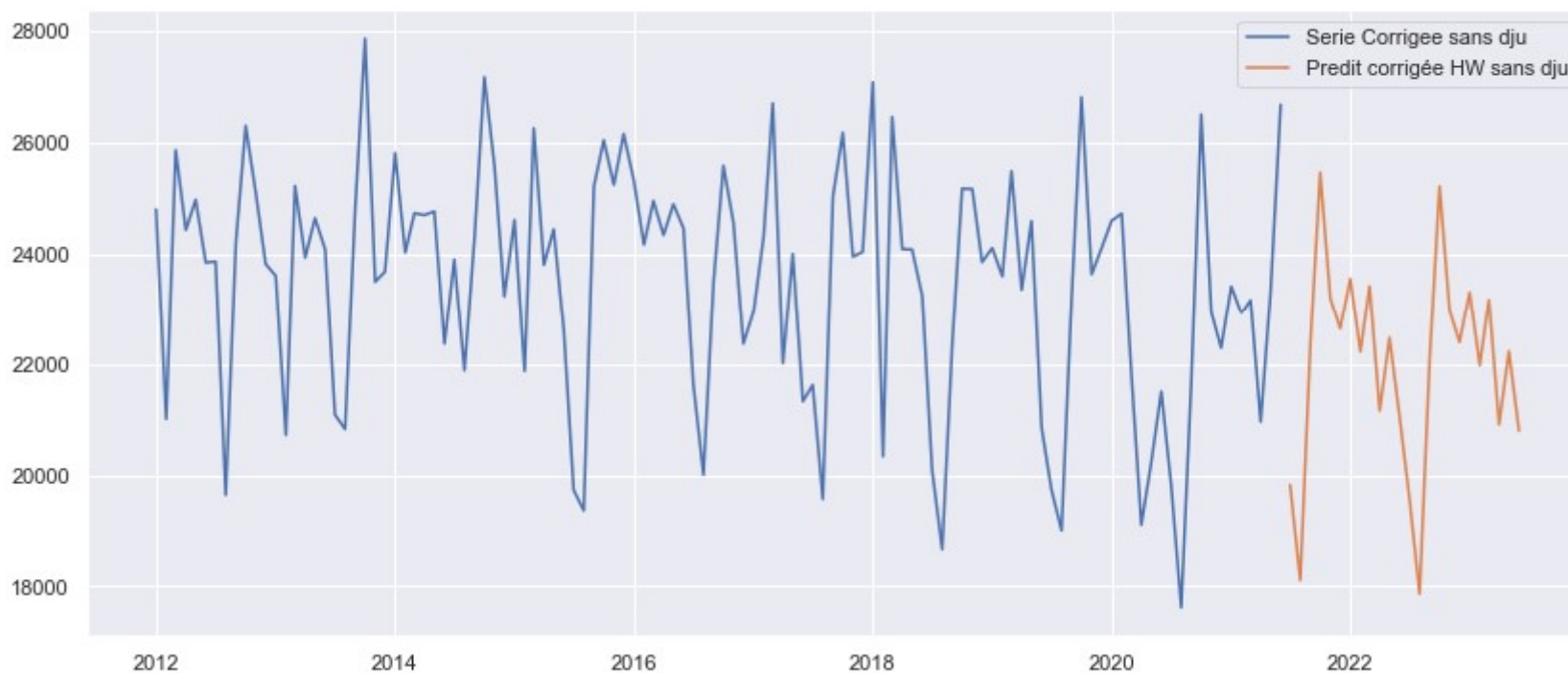


Figure 14 : Prédiction de la consommation électrique corrigée sans DJU pour les 2 prochaines années par le modèle Hotwinter

Modèle prédictif Hotwinter pour l'année à venir

27

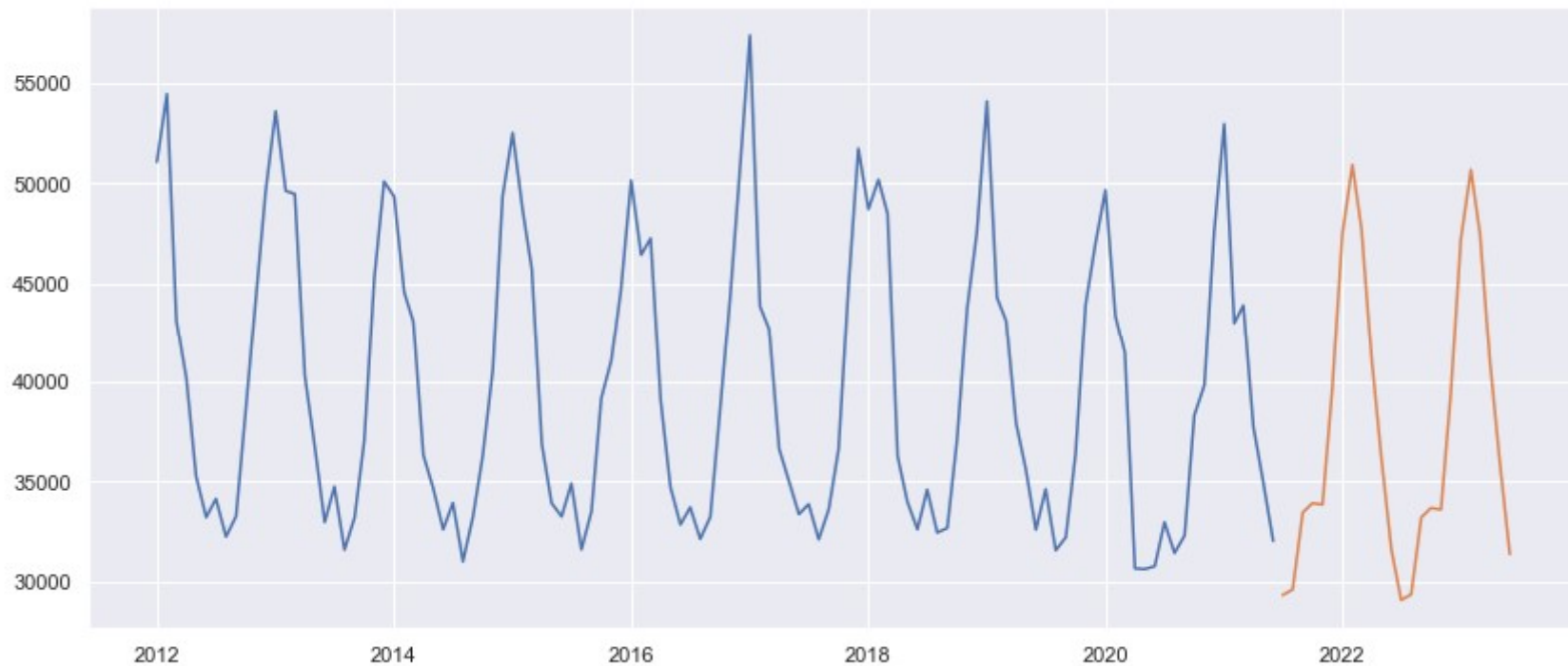


Figure 15 : Prédiction de la consommation électrique avec le DJU moyen pour les 2 prochaines années par le modèle Hotwinter

Modèle prédictif Hotwinter pour l'année à venir

28

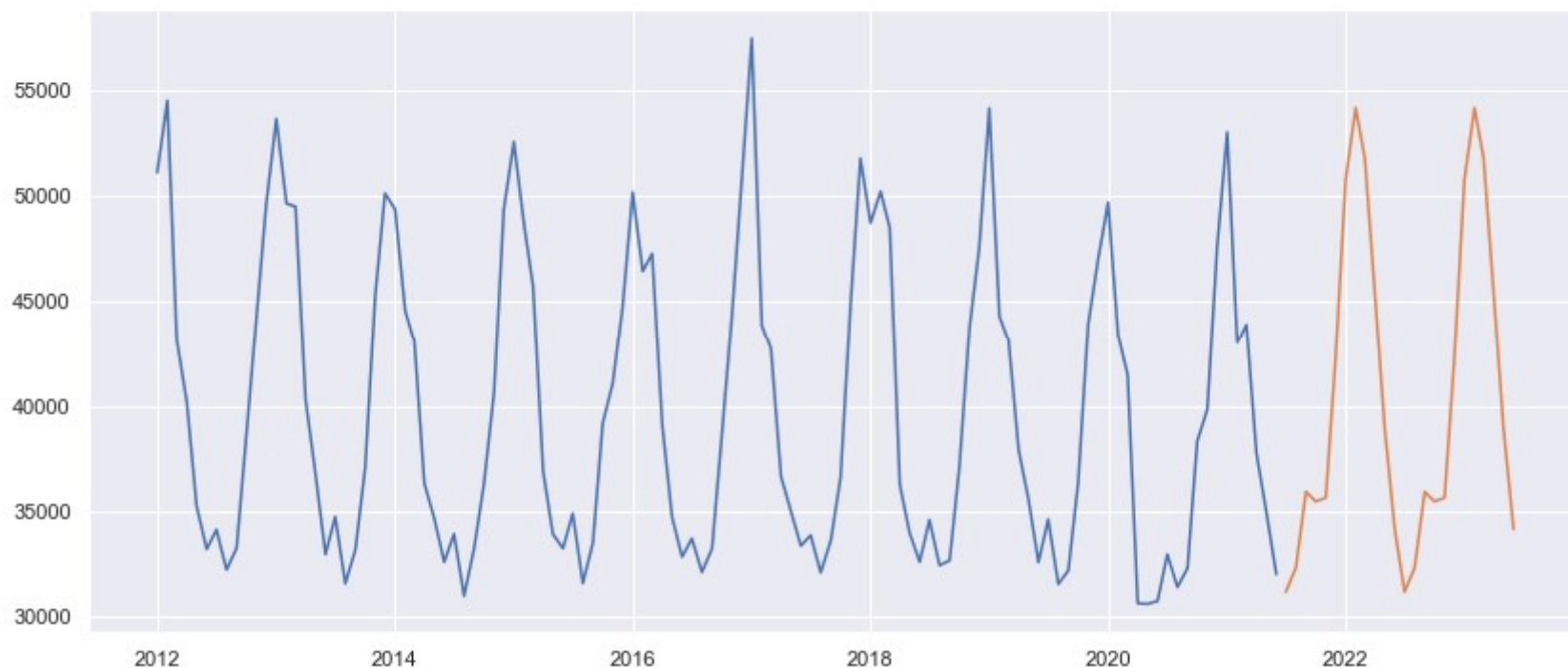


Figure 16 : Prédiction de la consommation électrique avec le DJU moyen pour les 2 prochaines années par le modèle Hotwinter sans tendance à la baisse

Conclusion

29

- Hotwinter

```
MAE=np.abs(x_a_prevoir['corrige']-x_a_prevoir['prediction']).mean()  
MAE  
1077.7032
```

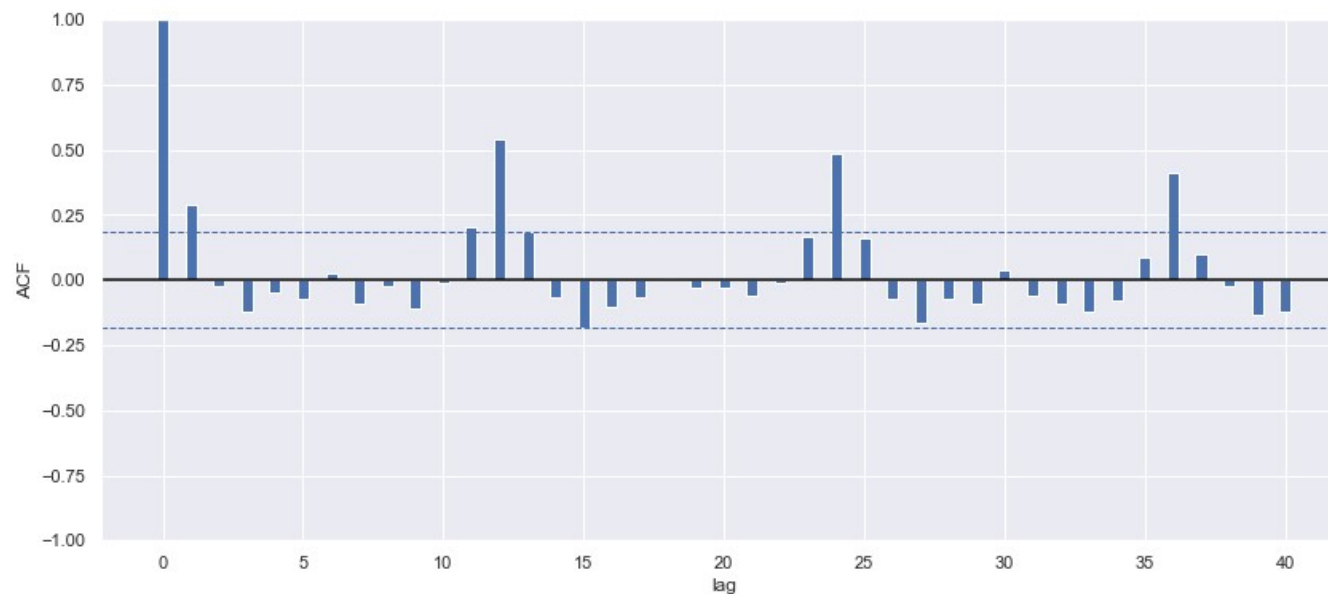
- Les variations de consommations électrique sont en grande partie expliquées par le DJU et par un effet de saisonnalité

ACF

30

- Décroissance lente au niveau des lags 12, on n'effectue pas de différentiation en tendance donc $d=0$ mais une différenciation en saisonnalité donc $D=1$

```
adfuller, analyse de corrélation de la série  
0.6699528391819632  
kpss, tendance stationnaire  
0.03954403635661572  
p-values < 5%, donc la série est stationnaire
```

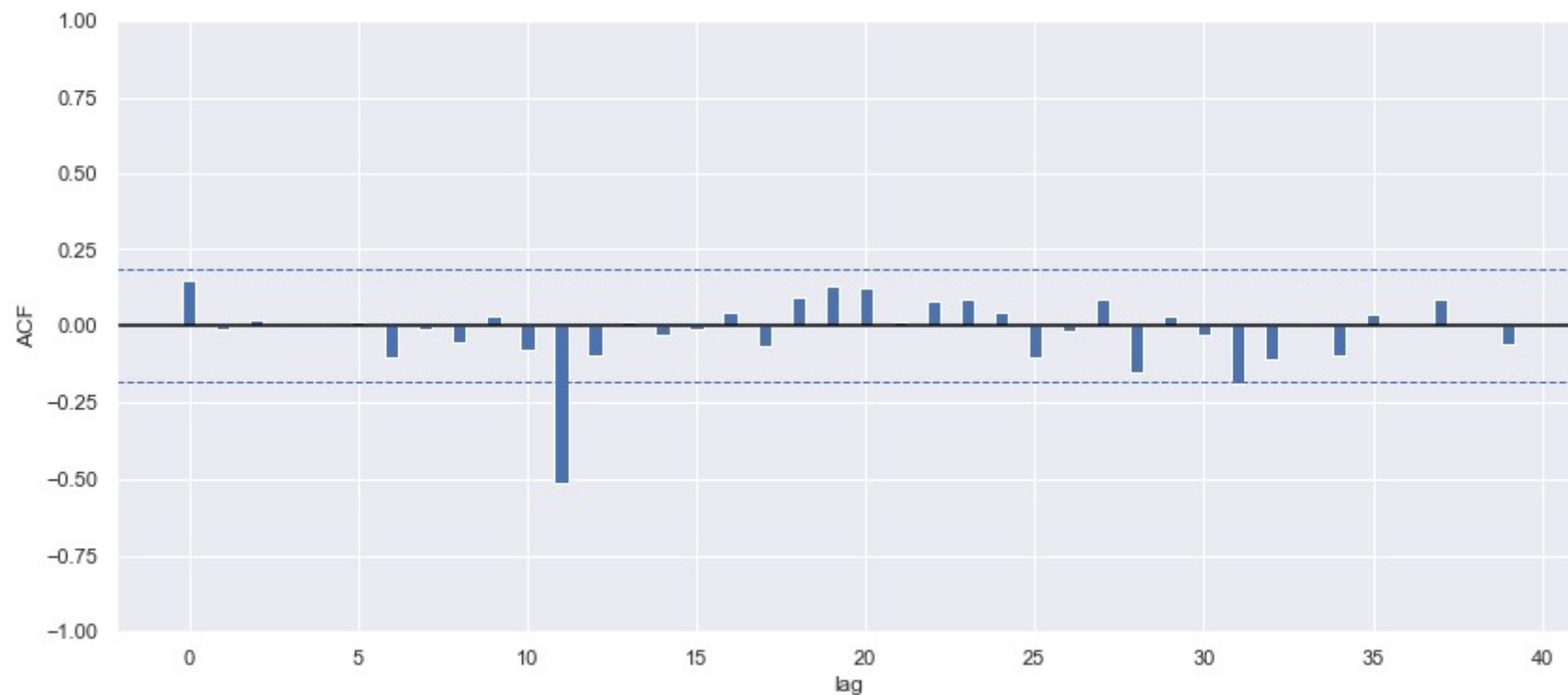


PACF

31

```
# use differenciation (i - B)

serie_corr_df_diff = serie_corr_df['corrige'] - serie_corr_df['corrige'].shift(12)
plot_sortie_acf(acf(np.asarray(serie_corr_df_diff[13:])), len(serie_corr_df_diff), pacf=True)
```



On regarde les lags saisonniers qui sortent du seuil de significativité sur le PACF
 $P = 1$