Get all Imports

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

Load Data

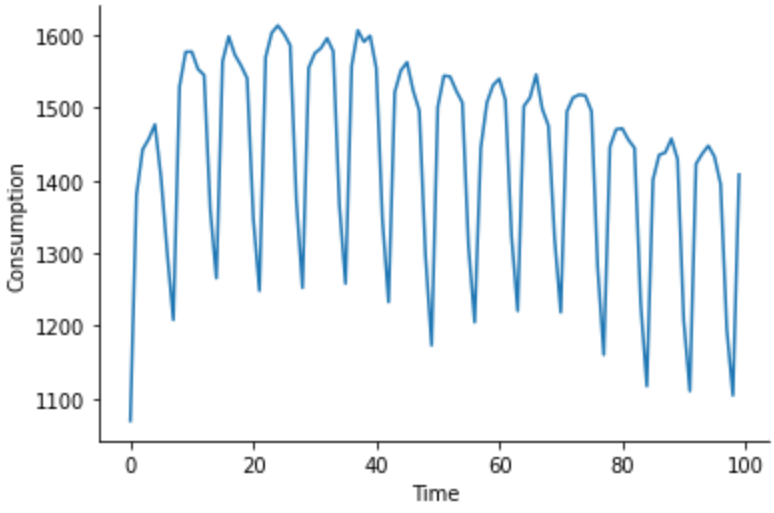
Data Description: We are going to predict the amount of power Consumption at time t based on previous time points $t-1, t-2 \ldots t-i$ where i is a user defined number

In [ ]:
```
# Load the data
df = pd.read_csv("auto-regressive-data.csv")
display(df.head())

# We are going to use Consmption as our target variable.
data = df["Consumption"].values

# Check out the data
sns.lineplot(x=np.arange(len(data[:100])), y=data[:100])
plt.xlabel("Time")
plt.ylabel("Consumption")
sns.despine()
plt.show()
```

	Date	Consumption	Wind	Solar	Wind+Solar
0	2006-01-01	1069.184	NaN	NaN	NaN
1	2006-01-02	1380.521	NaN	NaN	NaN
2	2006-01-03	1442.533	NaN	NaN	NaN
3	2006-01-04	1457.217	NaN	NaN	NaN
4	2006-01-05	1477.131	NaN	NaN	NaN



The following function creates lagged data

In [ ]:
```
def create_lag(data, lag):
    X, Y = list(), list()
    for i in range(len(data)-lag):
        X.append(data[i:i+lag])
        Y.append(data[i+lag])
    return np.asarray(X), np.asarray(Y)
```

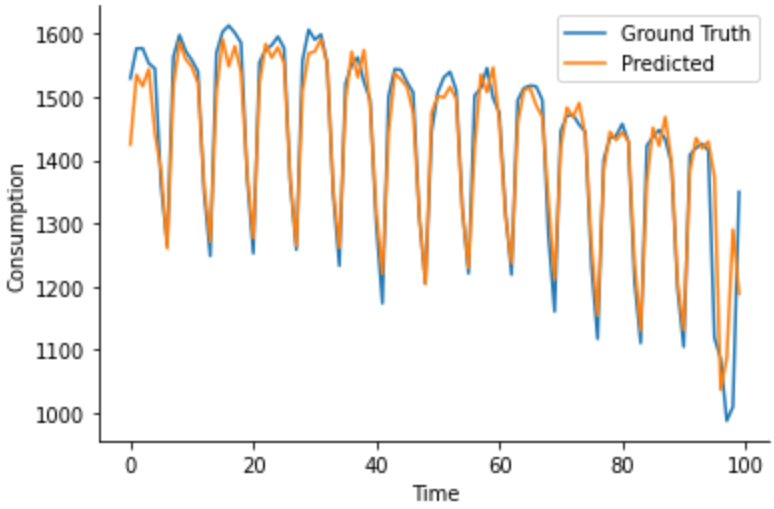
Finally, generate lagged data and apply Regression

In [ ]:
```
# Generate lagged data
X, Y = create_lag(data, 8)

# Fit a linear regression model
model = LinearRegression()
model.fit(X, Y)
preds = model.predict(X)
print("Root Mean Squared Error:", np.sqrt(np.sum(np.square(preds-Y))/len(Y)))

# Plot the predicted points
sns.lineplot(x=np.arange(len(Y[:100])), y=Y[:100], label="Ground Truth")
sns.lineplot(x=np.arange(len(preds[:100])), y=preds[:100], label="Predicted")
plt.xlabel("Time")
plt.ylabel("Consumption")
sns.despine()
plt.show()
```

```
Root Mean Squared Error: 70.08355832517167
```



What happens when we take very small amount of lag

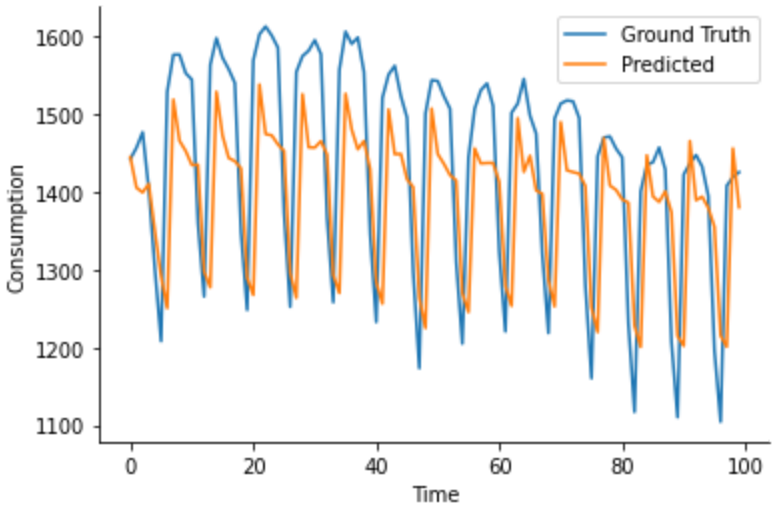
In [ ]:
```
# Generate lagged data
X, Y = create_lag(data, 2)

# Fit a linear regression model
model = LinearRegression()
model.fit(X, Y)
preds = model.predict(X)
```

```
print("Root Mean Squared Error:", np.sqrt(np.sum(np.square(preds-Y))/len(Y)))

# Plot the predicted points
sns.lineplot(x=np.arange(len(Y[:100])), y=Y[:100], label="Ground Truth")
sns.lineplot(x=np.arange(len(preds[:100])), y=preds[:100], label="Predicted")
plt.xlabel("Time")
plt.ylabel("Consumption")
sns.despine()
plt.show()
```

```
Root Mean Squared Error: 128.29021662454758
```



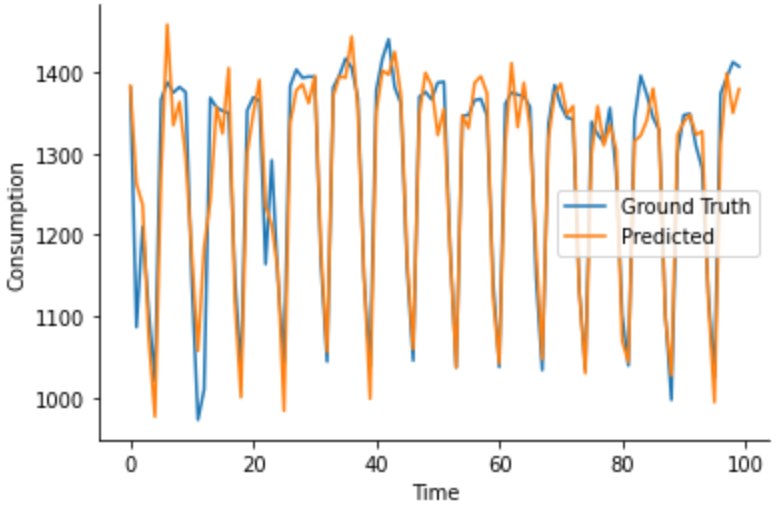
What happens when we take large amount of lag

In [ ]:
```
# Generate lagged data
X, Y = create_lag(data, 500)

# Fit a linear regression model
model = LinearRegression()
model.fit(X, Y)
preds = model.predict(X)
print("Root Mean Squared Error:", np.sqrt(np.sum(np.square(preds-Y))/len(Y)))

# Plot the predicted points
sns.lineplot(x=np.arange(len(Y[:100])), y=Y[:100], label="Ground Truth")
sns.lineplot(x=np.arange(len(preds[:100])), y=preds[:100], label="Predicted")
plt.xlabel("Time")
plt.ylabel("Consumption")
sns.despine()
plt.show()
```

```
Root Mean Squared Error: 44.53139065394836
```


Consumption
Time
Ground Truth
Predicted
1400
1300
1200
1100
1000
0
20
40
60
80
100

In [ ]: