Assignment 2. Time series:

ARMA, ARIMA, SARIMA application for forecast

A01232580 - Víctor Benito García Rocha

A mining transnational company wants to invest in Africa. It wants to know how the future Price of “cobalt” in USD/Dollar will rise for the next (2 years) 24 months.

- **Import the libraries**

In [ ]:
```
import pandas as pd # Data manipulation
import numpy as np # Data manipulation
import statistics as stats # Statistics
import matplotlib.pyplot as plt # Plotting
from statsmodels.tsa.stattools import adfuller, acf, pacf # Stationarity tests
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf # ACF and PACF
from statsmodels.tsa.arima.model import ARIMA # ARIMA model
from statsmodels.tsa.arima.model import ARIMAResults # ARIMA results
from statsmodels.tsa.arima_process import ArmaProcess # ARMA process
from statsmodels.tsa.statespace.sarimax import SARIMAX # SARIMA model
import matplotlib.pyplot as plt # Plotting
import statsmodels.api as sm # Time series analysis
%matplotlib inline # Display plots inline
import matplotlib.pyplot as plt # Plotting
```

```
UsageError: unrecognized arguments: # Display plots inline
```

- **Import the dataset**

In [ ]:
```
cobalt = pd.read_csv("cobalt.csv", sep=';', header=1, usecols=[0, 1], names=['date'
cobalt.head()
```

Out[ ]:

	date	Cobalt_USD/ton
0	2000M1	14437.50
1	2000M2	14093.75
2	2000M3	15225.00
3	2000M4	16218.75
4	2000M5	16543.48

- **Transform the dataset**

In [ ]:
```
# Transform the date column to datetime
cobalt['date'] = pd.to_datetime(cobalt['date'], format='%YM%m')
cobalt.head()
```

Out[ ]:

	date	Cobalt_USD/ton
0	2000-01-01	14437.50
1	2000-02-01	14093.75
2	2000-03-01	15225.00
3	2000-04-01	16218.75
4	2000-05-01	16543.48

In [ ]:
```
# Set the date column as index
cobalt.set_index('date', inplace=True)
cobalt.head()
```

Out[ ]:

	Cobalt_USD/ton
date	
2000-01-01	14437.50
2000-02-01	14093.75
2000-03-01	15225.00
2000-04-01	16218.75
2000-05-01	16543.48

In [ ]:
```
# Select the cobalt column as a time series
ts = cobalt['Cobalt_USD/ton']
cobalt['Cobalt_USD/ton'].describe()
```

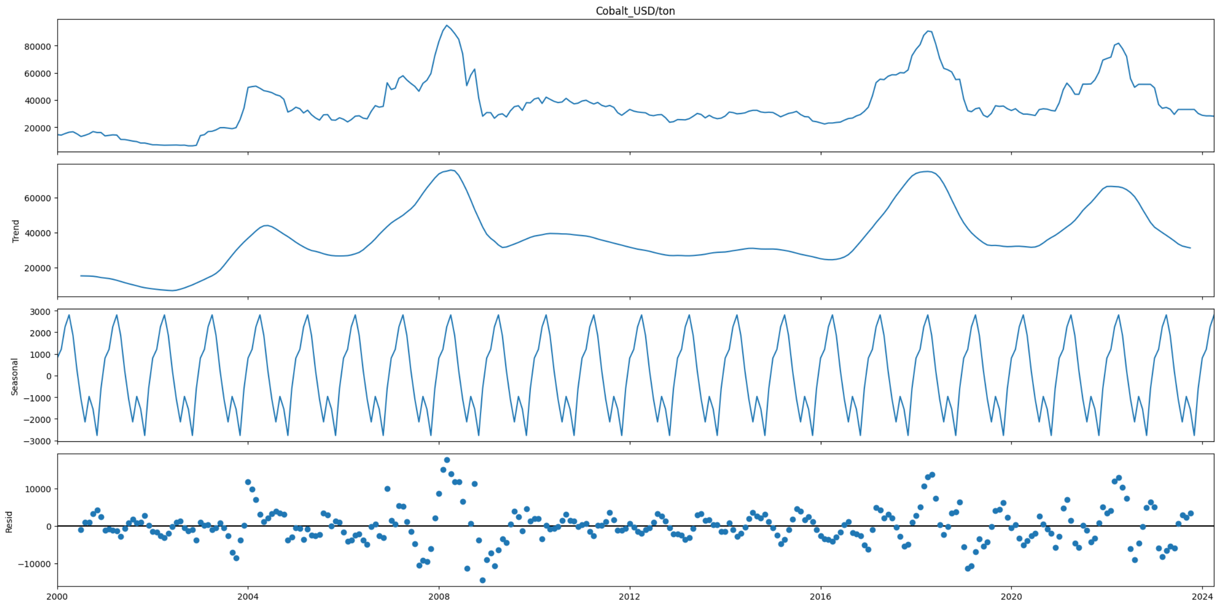
Out[ ]:
```
count      292.000000
mean     36214.245651
std      18423.669056
min       6185.710000
25%      26580.090000
50%      31734.910000
75%      45589.885000
max      95022.920000
Name: Cobalt_USD/ton, dtype: float64
```

In [ ]:
```
# See if there are any missing values
cobalt.isnull().sum()
```

Out[ ]:
```
Cobalt_USD/ton    0
dtype: int64
```

- **General pot**

```
In [ ]: # Plot variable, Trend, Seasonal and Resid
        dec = sm.tsa.seasonal_decompose(cobalt['Cobalt_USD/ton'], period = 12, model = 'add
        plt.show()
```



- **# Perform a Dickey-Fuller test**

The p-value is less than 0.05, so we can reject the null hypothesis that the time series is non-stationary (so it is stationary).

```
In [ ]: result = adfuller(cobalt['Cobalt_USD/ton'], autolag='AIC')
        output = pd.Series(result[0:4], index=['Test Statistic', 'p-value', '#Lags Used', '
        for key, value in result[4].items():
            output['Critical Value (%s)' % key] = value
        print(output)
```

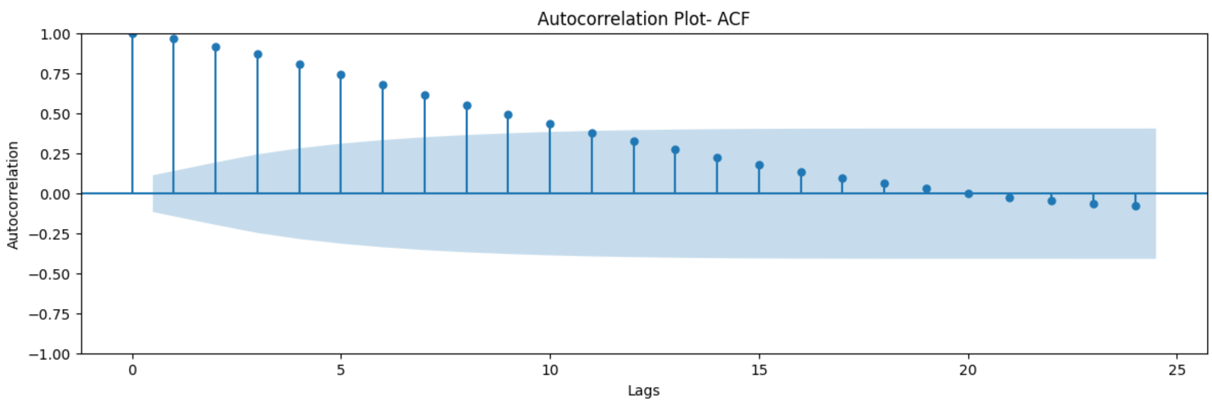
```
Test Statistic                  -3.477597
p-value                          0.008581
#Lags Used                       3.000000
Number of Observations Used    288.000000
Critical Value (1%)             -3.453262
Critical Value (5%)             -2.871628
Critical Value (10%)            -2.572146
dtype: float64
```

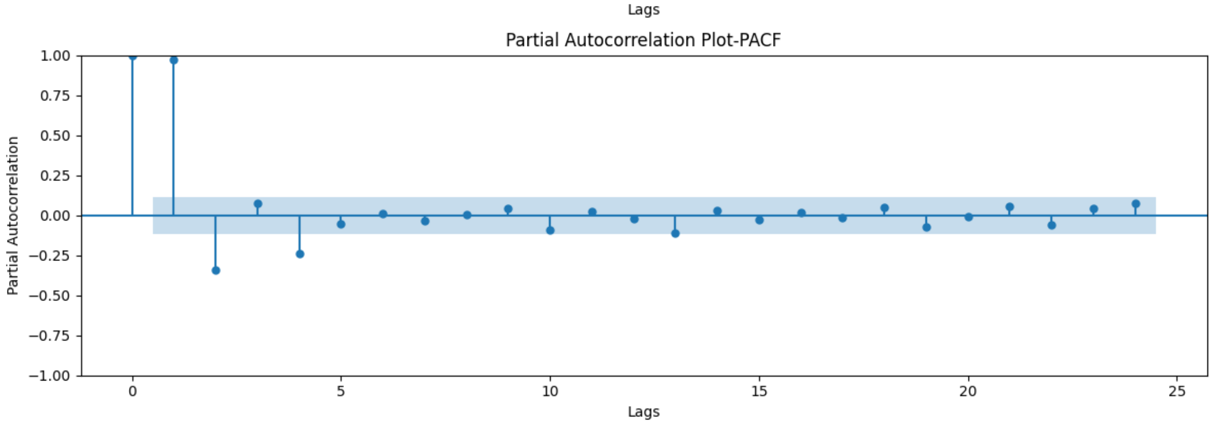
- **Autocorrelation plots**

The value from the ACF should be 20 (which is the number of lags for 2 years), so we can use q=20. And from PACF 10 (which is the number of lags for 1 year), so we can use p=10.

```
In [ ]: # Select column
        price = cobalt['Cobalt_USD/ton']

        # Create subgraph figures
```

```
fig, axes = plt.subplots(2, 1, figsize=(12, 8))

# Plot autocorrelation (MA(ACF))
plot_acf(price, ax = axes[0], lags= 24) # 24 lags for 2 years
axes[0].set_xlabel('Lags')
axes[0].set_ylabel('Autocorrelation')
axes[0].set_title('Autocorrelation Plot- ACF')

# Plot partial autocorrelation (AR(PACF))
plot_pacf(price, ax=axes[1], lags = 24)
axes[1].set_xlabel('Lags')
axes[1].set_ylabel('Partial Autocorrelation')
axes[1].set_title('Partial Autocorrelation Plot-PACF')

plt.tight_layout() # Adjust spaces
plt.show()
```





- **Diferentiation**

From the ADF test, we can see that the first and second differenciation are stationary. So we can use d=1 and D=1.

In [ ]:
```
# Function to apply adfuller
def adf_test(df):
    result = adfuller(df)
    print('Estadístico ADF:', result[0])
    print('Valor p:', result[1])
    print('Valores Críticos:')
    for key, value in result[4].items():
        print(f'   {key}: {value}')
```

```
# First differenciation
ts_diff1 = ts.diff().dropna()
print("\nADF test for the first differenciation:")
adf_test(ts_diff1)

# Second sifferenciation
ts_diff2 = ts_diff1.diff().dropna()
print("\nADF test for the second differenciation:")
adf_test(ts_diff2)
```

```
ADF test for the first differenciation:
Estadístico ADF: -7.116670970605816
Valor p: 3.815089725333254e-10
Valores Críticos:
   1%: -3.453261605529366
   5%: -2.87162848654246
   10%: -2.5721455328896603

ADF test for the second differenciation:
Estadístico ADF: -8.40006175932168
Valor p: 2.2439421553910847e-13
Valores Críticos:
   1%: -3.4540076534999957
   5%: -2.8719557347997178
   10%: -2.5723200648758366
```

a) Estimate ARMA, ARIMA and SARIMA models.

- **ARMA Model**

In [ ]:

```
# Fit the model
arma = ARIMA(ts, order=(10, 0, 20))

# Fit the model
arma_fit = arma.fit()

# Print the summary
print(arma_fit.summary())
```

```
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provi
ded, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provi
ded, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provi
ded, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimization fail
ed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:         Cobalt_USD/ton   No. Observations:                  292
Model:                ARIMA(10, 0, 20)  Log Likelihood               -2808.764
Date:                Tue, 16 Jul 2024   AIC                           5681.529
Time:                        22:39:41   BIC                           5799.185
Sample:                    01-01-2000   HQIC                          5728.657
                         - 04-01-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       3.621e+04   5891.620      6.147      0.000    2.47e+04    4.78e+04
ar.L1          0.7933      0.847      0.937      0.349      -0.867       2.453
ar.L2          0.5380      0.842      0.639      0.523      -1.112       2.188
ar.L3         -0.0868      0.449     -0.193      0.847      -0.967       0.793
ar.L4         -0.5175      0.433     -1.195      0.232      -1.366       0.331
ar.L5         -0.1730      0.294     -0.589      0.556      -0.749       0.403
ar.L6          0.7514      0.352      2.134      0.033       0.061       1.441
ar.L7          0.2129      0.683      0.312      0.755      -1.127       1.553
ar.L8         -0.4772      0.463     -1.030      0.303      -1.386       0.431
ar.L9         -0.5121      0.437     -1.171      0.241      -1.369       0.345
ar.L10         0.3617      0.393      0.920      0.358      -0.409       1.132
ma.L1          0.5912      0.837      0.706      0.480      -1.049       2.232
ma.L2         -0.3513      0.595     -0.590      0.555      -1.517       0.815
ma.L3         -0.3296      0.360     -0.915      0.360      -1.035       0.376
ma.L4          0.3791      0.260      1.461      0.144      -0.129       0.888
ma.L5          0.5831      0.286      2.039      0.041       0.022       1.144
ma.L6         -0.3746      0.458     -0.818      0.413      -1.273       0.523
ma.L7         -0.8398      0.364     -2.309      0.021      -1.553      -0.127
ma.L8         -0.3455      0.565     -0.612      0.541      -1.453       0.762
ma.L9          0.4714      0.510      0.924      0.355      -0.528       1.471
ma.L10         0.1557      0.198      0.786      0.432      -0.233       0.544
ma.L11        -0.0821      0.169     -0.487      0.627      -0.413       0.249
ma.L12         0.1359      0.183      0.742      0.458      -0.223       0.495
ma.L13         0.2290      0.215      1.065      0.287      -0.192       0.650
ma.L14         0.1862      0.252      0.739      0.460      -0.308       0.680
ma.L15         0.1634      0.275      0.594      0.552      -0.375       0.702
ma.L16         0.1380      0.250      0.553      0.580      -0.351       0.627
ma.L17        -0.0226      0.200     -0.113      0.910      -0.414       0.369
ma.L18         0.0273      0.170      0.160      0.873      -0.306       0.361
ma.L19         0.1814      0.161      1.125      0.261      -0.135       0.497
ma.L20         0.0420      0.212      0.198      0.843      -0.374       0.458
sigma2      1.519e+07      1.606   9.46e+06      0.000    1.52e+07    1.52e+07
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):               175.02
Prob(Q):                              0.90   Prob(JB):                         0.00
Heteroskedasticity (H):               1.05   Skew:                             0.12
Prob(H) (two-sided):                  0.81   Kurtosis:                         6.78
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-ste
p).
[2] Covariance matrix is singular or near-singular, with condition number 5.09e+22.
Standard errors may be unstable.
```

The AIC and BIC show that the model is not good. The p-values are greater than 0.05, so the model is not significant.

- **ARIMA Model**

In [ ]:
```
# Fit the model
arima = ARIMA(ts, order=(10, 1, 20))

# Fit the model
arima_fit = arima.fit()

# Print the summary
print(arima_fit.summary())
```

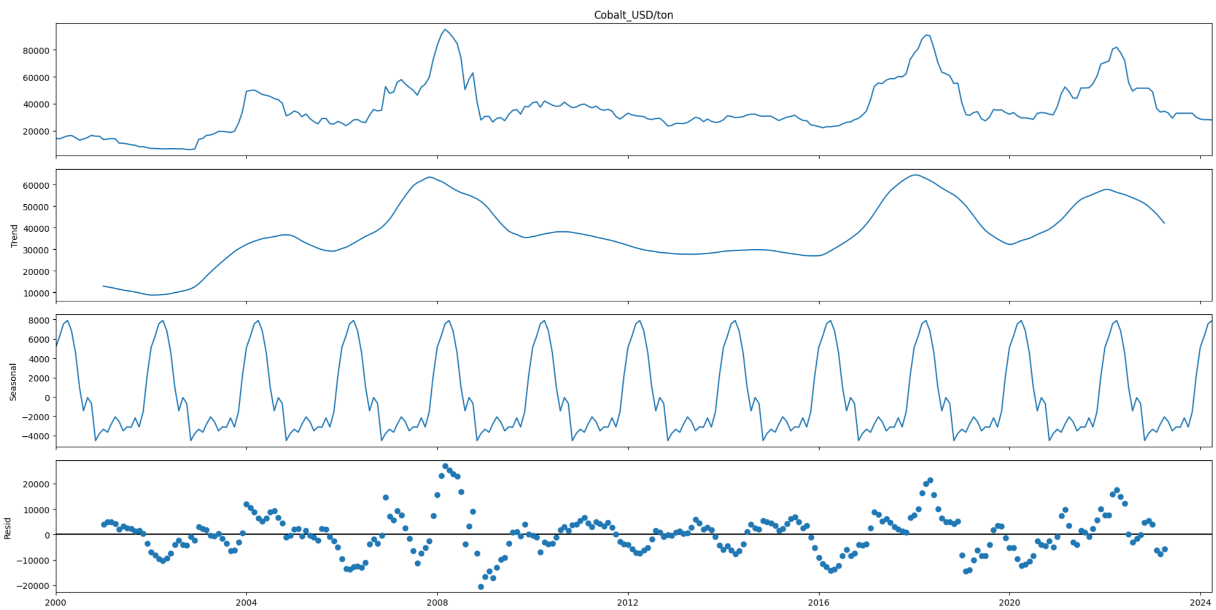
```
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provi
ded, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provi
ded, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provi
ded, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\sta
tsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimization fail
ed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:         Cobalt_USD/ton   No. Observations:                  292
Model:                ARIMA(10, 1, 20)  Log Likelihood               -2804.603
Date:                Tue, 16 Jul 2024   AIC                           5671.207
Time:                        22:39:48   BIC                           5785.080
Sample:                    01-01-2000   HQIC                          5716.825
                         - 04-01-2024
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3823      1.915      0.200      0.842      -3.372       4.137
ar.L2          0.6660      0.983      0.678      0.498      -1.260       2.592
ar.L3          0.0527      0.655      0.080      0.936      -1.232       1.337
ar.L4         -0.2185      0.727     -0.300      0.764      -1.644       1.207
ar.L5          0.0723      0.238      0.304      0.761      -0.393       0.538
ar.L6          0.3414      0.326      1.046      0.296      -0.298       0.981
ar.L7          0.1491      0.898      0.166      0.868      -1.612       1.910
ar.L8         -0.6406      1.050     -0.610      0.542      -2.698       1.417
ar.L9         -0.4673      0.444     -1.054      0.292      -1.337       0.402
ar.L10         0.5458      1.303      0.419      0.675      -2.009       3.101
ma.L1          0.0341      1.907      0.018      0.986      -3.703       3.771
ma.L2         -0.8657      1.763     -0.491      0.623      -4.322       2.590
ma.L3         -0.2324      0.416     -0.558      0.577      -1.048       0.584
ma.L4          0.3436      0.888      0.387      0.699      -1.397       2.085
ma.L5         -0.0945      0.256     -0.370      0.712      -0.596       0.407
ma.L6         -0.6158      0.329     -1.874      0.061      -1.260       0.028
ma.L7         -0.2744      1.419     -0.193      0.847      -3.056       2.507
ma.L8          0.5979      1.733      0.345      0.730      -2.799       3.995
ma.L9          0.7543      0.366      2.063      0.039       0.038       1.471
ma.L10        -0.4687      1.333     -0.352      0.725      -3.082       2.145
ma.L11        -0.2226      0.285     -0.782      0.434      -0.781       0.335
ma.L12         0.2669      0.296      0.900      0.368      -0.314       0.848
ma.L13         0.0751      0.395      0.190      0.849      -0.698       0.848
ma.L14        -0.1778      0.485     -0.367      0.714      -1.129       0.773
ma.L15        -0.0136      0.175     -0.078      0.938      -0.356       0.329
ma.L16         0.0272      0.136      0.200      0.842      -0.240       0.294
ma.L17        -0.1479      0.169     -0.875      0.381      -0.479       0.183
ma.L18        -0.0415      0.263     -0.158      0.874      -0.556       0.473
ma.L19         0.1053      0.363      0.290      0.772      -0.607       0.817
ma.L20        -0.0164      0.173     -0.095      0.924      -0.355       0.322
sigma2       1.58e+07    1.2e-06   1.32e+13      0.000    1.58e+07    1.58e+07
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):               297.37
Prob(Q):                              0.89   Prob(JB):                         0.00
Heteroskedasticity (H):               1.04   Skew:                            -0.06
Prob(H) (two-sided):                  0.85   Kurtosis:                         7.95
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-ste
p).
[2] Covariance matrix is singular or near-singular, with condition number 1.58e+29.
Standard errors may be unstable.
```

Again, the AIC and BIC show that the model is not good. The p-values are greater than 0.05, so the model is not significant. We require to use the SARIMA model.

- **SARIMA Model**

In [ ]:
```python
# Adjust SARIMA model
from pylab import rcParams
import statsmodels.api as sm
rcParams['figure.figsize'] = 20, 10
decomposition = sm.tsa.seasonal_decompose(ts, model='additive', period=24)
fig = decomposition.plot()
plt.show()
```



In [ ]:
```python
sarima = sm.tsa.SARIMAX(ts, order=(10, 1, 10), seasonal_order=(2, 1, 5, 12))

# Train the model
sarima_fit = sarima.fit(disp = False)

# Print the summary
print(sarima_fit.summary())
```

```
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\Administrador\AppData\Local\Programs\Python\Python312\Lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

```
                                               SARIMAX Results
=====================================================================================
======================
Dep. Variable:                                    Cobalt_USD/ton   No. Observation
s:                   292
Model:             SARIMAX(10, 1, 10)x(2, 1, [1, 2, 3, 4, 5], 12)   Log Likelihood
-2727.145
Date:                                           Tue, 16 Jul 2024   AIC
5510.289
Time:                                                   22:42:46   BIC
5611.963
Sample:                                               01-01-2000   HQIC
5551.075
                                                    - 04-01-2024
Covariance Type:                                             opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1194      0.923      0.129      0.897      -1.690       1.928
ar.L2          0.0835      1.234      0.068      0.946      -2.334       2.501
ar.L3          0.6610      0.602      1.098      0.272      -0.519       1.841
ar.L4         -0.0059      0.830     -0.007      0.994      -1.632       1.621
ar.L5          0.4206      0.823      0.511      0.609      -1.193       2.034
ar.L6         -0.2563      0.878     -0.292      0.770      -1.978       1.465
ar.L7         -0.2019      1.149     -0.176      0.861      -2.453       2.050
ar.L8         -0.5846      0.349     -1.677      0.094      -1.268       0.099
ar.L9          0.3057      0.754      0.406      0.685      -1.172       1.783
ar.L10         0.2190      0.805      0.272      0.786      -1.360       1.798
ma.L1          0.2148      0.912      0.235      0.814      -1.573       2.003
ma.L2         -0.1975      1.321     -0.149      0.881      -2.787       2.392
ma.L3         -0.6038      0.889     -0.679      0.497      -2.346       1.138
ma.L4         -0.0383      0.804     -0.048      0.962      -1.614       1.538
ma.L5         -0.3899      0.745     -0.523      0.601      -1.850       1.071
ma.L6         -0.0068      0.891     -0.008      0.994      -1.752       1.739
ma.L7          0.2533      1.188      0.213      0.831      -2.075       2.581
ma.L8          0.5003      0.447      1.120      0.263      -0.375       1.376
ma.L9         -0.1743      0.625     -0.279      0.780      -1.399       1.051
ma.L10        -0.4122      0.580     -0.710      0.477      -1.549       0.725
ar.S.L12      -0.0686      0.922     -0.074      0.941      -1.877       1.739
ar.S.L24      -0.2914      0.865     -0.337      0.736      -1.987       1.404
ma.S.L12      -0.7572      0.905     -0.836      0.403      -2.532       1.018
ma.S.L24       0.0622      0.965      0.064      0.949      -1.828       1.953
ma.S.L36      -0.1748      0.807     -0.216      0.829      -1.757       1.408
ma.S.L48       0.2252      0.261      0.863      0.388      -0.286       0.737
ma.S.L60      -0.1720      0.304     -0.567      0.571      -0.767       0.423
sigma2      2.953e+07   2.06e-07   1.43e+14      0.000    2.95e+07    2.95e+07
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):               263.03
Prob(Q):                              0.91   Prob(JB):                         0.00
Heteroskedasticity (H):               0.88   Skew:                            -0.74
Prob(H) (two-sided):                  0.54   Kurtosis:                         7.52
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-ste
p).
```

```
[2] Covariance matrix is singular or near-singular, with condition number 4.68e+30.
Standard errors may be unstable.
```

We accomplish our goal of having the lowest AIC and BIC values.

b) Choose which one is the best to make to forecast (AIC, BIC test).

The SARIMA model has the lowest AIC and BIC values, which means that it is the best model to make the forecast.

The reason to choose the lowest ones is because they penalize the number of parameters in the model.

- **ARMA:**

--- **AIC 5681.529**

--- **BIC 5799.185**

- **ARIMA:**

--- **AIC 5671.207**

--- **BIC 5785.080**

- **SARIMA:**

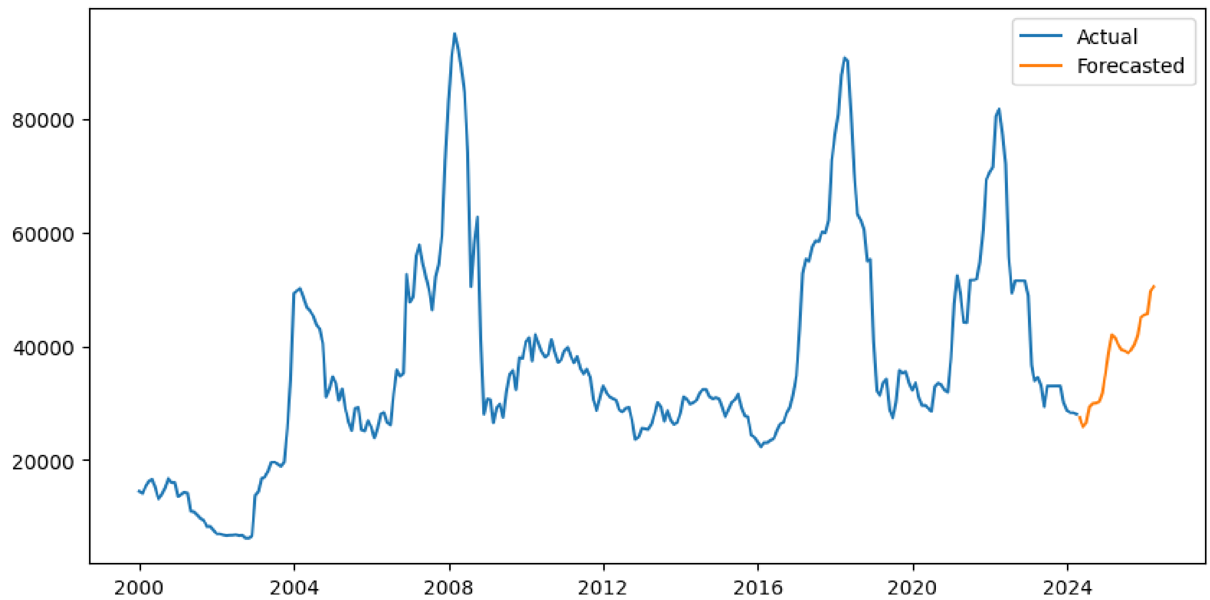
--- **AIC 5510.289**

--- **BIC 5611.963**

c) Visualize the forecast and interpret results

The SARIMA model effectively captures the underlying patterns in the cobalt price data by incorporating seasonal components, thereby achieving a good fit as shown by the alignment between actual and forecasted values in the graph, despite some parameter estimates not being statistically significant.

In [ ]:
```
# Forecast the next 24 months
forecast = sarima_fit.forecast(steps=24)

# Plot the forecast
plt.figure(figsize=(10, 5)) # Plot size
plt.plot(ts, label='Actual')
plt.plot(forecast, label='Forecasted')
plt.legend()
plt.show()
```


Actual
Forecasted
80000
60000
40000
20000
2000
2004
2008
2012
2016
2020
2024