In [2]:

```
import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.seasonal import seasonal_decompose

from pmdarima import auto_arima
```

In [3]:

```
from sklearn.metrics import mean_absolute_percentage_error
```

In [5]:

```
df=pd.read_csv('co2_mm_mlo_2022.csv',sep=',',comment='#',index_col=False)
df.index = pd.to_datetime(df[['year','month']].apply(lambda x: f'{x[0]}-{x[1]}',axis
df
```

```
C:\anaconda3\lib\site-packages\pandas\util\_decorators.py:311: ParserWarning: Length
of header or names does not match length of data. This leads to a loss of data with
index_col=False.
  return func(*args, **kwargs)
```

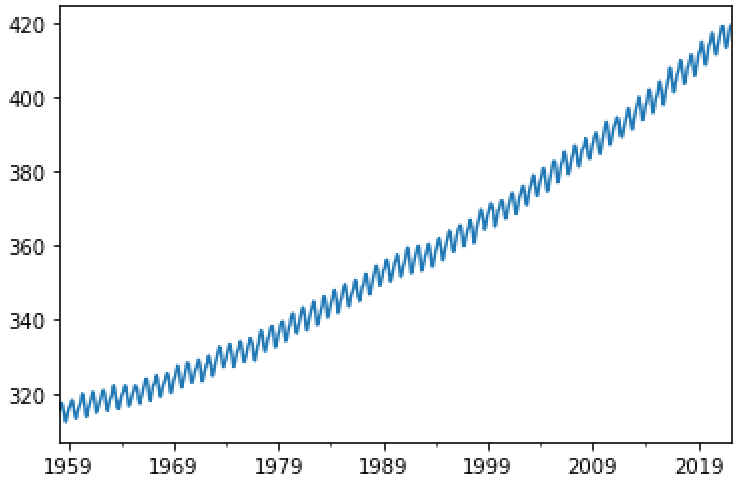
Out[5]:

	year	month	decimal date	average	interpolated	trend	ndays
1958-03-01	1958	3	1958.2027	315.70	314.43	-1	-9.99
1958-04-01	1958	4	1958.2877	317.45	315.16	-1	-9.99
1958-05-01	1958	5	1958.3699	317.51	314.71	-1	-9.99
1958-06-01	1958	6	1958.4548	317.24	315.14	-1	-9.99
1958-07-01	1958	7	1958.5370	315.86	315.18	-1	-9.99
...	...	...	...	...	...	...	...
2021-11-01	2021	11	2021.8750	415.01	417.05	30	0.36
2021-12-01	2021	12	2021.9583	416.71	417.46	28	0.48
2022-01-01	2022	1	2022.0417	418.19	417.84	29	0.73
2022-02-01	2022	2	2022.1250	419.28	418.36	27	0.92
2022-03-01	2022	3	2022.2083	418.81	417.31	30	0.76

769 rows × 7 columns

In [6]:

```
df['average'].plot()
avg=df['average']
```

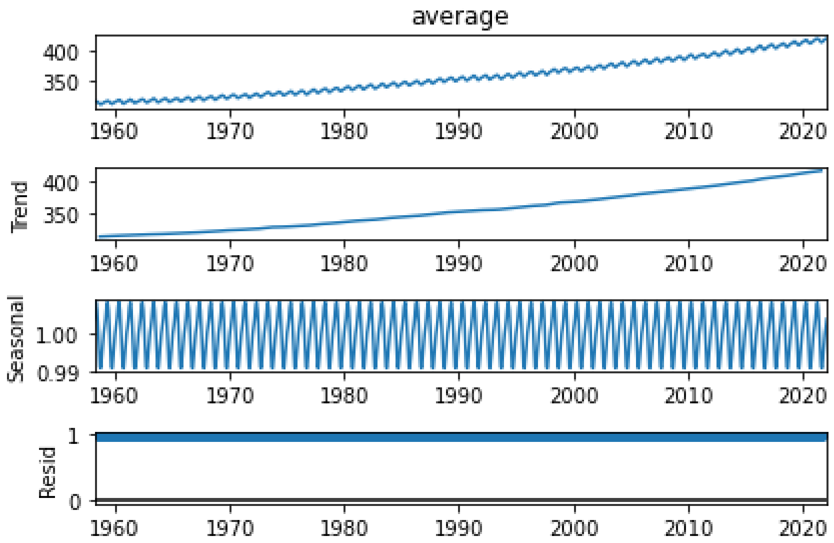


In [7]:

```
df.index.freq='MS'
```

In [8]:

```
seasonalDecomposition=seasonal_decompose(avg,model='mul')
seasonalDecomposition.plot();
```



In [9]:

```
auto_arima(avg, seasonal=True, m=12, trace=True).summary
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(1,0,1)[12] intercept   : AIC=488.228, Time=1.58 sec
 ARIMA(0,1,0)(0,0,0)[12] intercept   : AIC=2503.427, Time=0.01 sec
 ARIMA(1,1,0)(1,0,0)[12] intercept   : AIC=inf, Time=0.46 sec
 ARIMA(0,1,1)(0,0,1)[12] intercept   : AIC=1704.374, Time=0.28 sec
 ARIMA(0,1,0)(0,0,0)[12]             : AIC=2510.500, Time=0.02 sec
 ARIMA(2,1,2)(0,0,1)[12] intercept   : AIC=inf, Time=1.42 sec
 ARIMA(2,1,2)(1,0,0)[12] intercept   : AIC=807.253, Time=1.52 sec
 ARIMA(2,1,2)(2,0,1)[12] intercept   : AIC=533.818, Time=3.22 sec
 ARIMA(2,1,2)(1,0,2)[12] intercept   : AIC=550.119, Time=3.59 sec
 ARIMA(2,1,2)(0,0,0)[12] intercept   : AIC=1522.553, Time=0.45 sec
 ARIMA(2,1,2)(0,0,2)[12] intercept   : AIC=inf, Time=3.16 sec
 ARIMA(2,1,2)(2,0,0)[12] intercept   : AIC=640.143, Time=3.31 sec
 ARIMA(2,1,2)(2,0,2)[12] intercept   : AIC=514.365, Time=4.28 sec
 ARIMA(1,1,2)(1,0,1)[12] intercept   : AIC=493.404, Time=1.45 sec
C:\anaconda3\lib\site-packages\pmdarima\arima\_auto_solvers.py:522: ModelFitWarning:
Error fitting  ARIMA(2,1,1)(1,0,1)[12] intercept (if you do not want to see these wa
rnings, run with error_action="ignore").
Traceback:
```

```
Traceback (most recent call last):
  File "C:\anaconda3\lib\site-packages\pmdarima\arima\_auto_solvers.py", line 506, i
n _fit_candidate_model
    fit.fit(y, X=X, **fit_params)
  File "C:\anaconda3\lib\site-packages\pmdarima\arima\arima.py", line 597, in fit
    self._fit(y, X, **fit_args)
  File "C:\anaconda3\lib\site-packages\pmdarima\arima\arima.py", line 518, in _fit
    fit, self.arima_res_ = _fit_wrapper()
  File "C:\anaconda3\lib\site-packages\pmdarima\arima\arima.py", line 508, in _fit_w
rapper
    return arima, arima.fit(start_params=start_params,
  File "C:\anaconda3\lib\site-packages\statsmodels\tsa\statespace\mlemodel.py", line
690, in fit
    mlefit = super(MLEModel, self).fit(start_params, method=method,
  File "C:\anaconda3\lib\site-packages\statsmodels\base\model.py", line 519, in fit
    xopt, retvals, optim_settings = optimizer._fit(f, score, start_params,
  File "C:\anaconda3\lib\site-packages\statsmodels\base\optimizer.py", line 224, in
_fit
    xopt, retvals = func(objective, gradient, start_params, fargs, kwargs,
  File "C:\anaconda3\lib\site-packages\statsmodels\base\optimizer.py", line 629, in
_fit_lbfgs
    retvals = optimize.fmin_l_bfgs_b(func, start_params, maxiter=maxiter,
  File "C:\anaconda3\lib\site-packages\scipy\optimize\lbfgsb.py", line 197, in fmin_
l_bfgs_b
    res = _minimize_lbfgsb(fun, x0, args=args, jac=jac, bounds=bounds,
  File "C:\anaconda3\lib\site-packages\scipy\optimize\lbfgsb.py", line 360, in _mini
mize_lbfgsb
    f, g = func_and_grad(x)
  File "C:\anaconda3\lib\site-packages\scipy\optimize\_differentiable_functions.py",
line 267, in fun_and_grad
    self._update_fun()
  File "C:\anaconda3\lib\site-packages\scipy\optimize\_differentiable_functions.py",
line 233, in _update_fun
    self._update_fun_impl()
  File "C:\anaconda3\lib\site-packages\scipy\optimize\_differentiable_functions.py",
line 137, in update_fun
    self.f = fun_wrapped(self.x)
  File "C:\anaconda3\lib\site-packages\scipy\optimize\_differentiable_functions.py",
line 134, in fun_wrapped
    return fun(np.copy(x), *args)
  File "C:\anaconda3\lib\site-packages\statsmodels\base\model.py", line 501, in f
    return -self.loglike(params, *args) / nobs
  File "C:\anaconda3\lib\site-packages\statsmodels\tsa\statespace\mlemodel.py", line
925, in loglike
    loglike = self.ssm.loglike(complex_step=complex_step, **kwargs)
  File "C:\anaconda3\lib\site-packages\statsmodels\tsa\statespace\kalman_filter.py",
line 983, in loglike
    kfilter = self._filter(**kwargs)
  File "C:\anaconda3\lib\site-packages\statsmodels\tsa\statespace\kalman_filter.py",
line 903, in _filter
    self._initialize_state(prefix=prefix, complex_step=complex_step)
  File "C:\anaconda3\lib\site-packages\statsmodels\tsa\statespace\representation.p
y", line 985, in _initialize_state
    self._statespaces[prefix].initialize(self.initialization,
  File "statsmodels\tsa\statespace\_representation.pyx", line 1373, in statsmodels.t
sa.statespace._representation.dStatespace.initialize
  File "statsmodels\tsa\statespace\_representation.pyx", line 1362, in statsmodels.t
sa.statespace._representation.dStatespace.initialize
  File "statsmodels\tsa\statespace\_initialization.pyx", line 288, in statsmodels.ts
a.statespace._initialization.dInitialization.initialize
  File "statsmodels\tsa\statespace\_initialization.pyx", line 406, in statsmodels.ts
a.statespace._initialization.dInitialization.initialize_stationary_stationary_cov
  File "statsmodels\tsa\statespace\_tools.pyx", line 1206, in statsmodels.tsa.states
pace._tools._dsolve_discrete_lyapunov
```

```
numpy.linalg.LinAlgError: LU decomposition error.

  warnings.warn(warning_str, ModelFitWarning)
 ARIMA(2,1,1)(1,0,1)[12] intercept   : AIC=inf, Time=nan sec
 ARIMA(3,1,2)(1,0,1)[12] intercept   : AIC=497.047, Time=1.85 sec
 ARIMA(2,1,3)(1,0,1)[12] intercept   : AIC=484.439, Time=2.01 sec
 ARIMA(2,1,3)(0,0,1)[12] intercept   : AIC=1512.654, Time=0.96 sec
 ARIMA(2,1,3)(1,0,0)[12] intercept   : AIC=816.892, Time=1.63 sec
 ARIMA(2,1,3)(2,0,1)[12] intercept   : AIC=470.500, Time=3.83 sec
 ARIMA(2,1,3)(2,0,0)[12] intercept   : AIC=648.339, Time=3.44 sec
 ARIMA(2,1,3)(2,0,2)[12] intercept   : AIC=465.773, Time=4.66 sec
 ARIMA(2,1,3)(1,0,2)[12] intercept   : AIC=494.823, Time=4.60 sec
 ARIMA(1,1,3)(2,0,2)[12] intercept   : AIC=463.065, Time=4.59 sec
 ARIMA(1,1,3)(1,0,2)[12] intercept   : AIC=516.787, Time=3.43 sec
 ARIMA(1,1,3)(2,0,1)[12] intercept   : AIC=500.444, Time=3.04 sec
 ARIMA(1,1,3)(1,0,1)[12] intercept   : AIC=497.966, Time=1.68 sec
 ARIMA(0,1,3)(2,0,2)[12] intercept   : AIC=476.352, Time=4.02 sec
 ARIMA(1,1,2)(2,0,2)[12] intercept   : AIC=inf, Time=4.10 sec
 ARIMA(1,1,4)(2,0,2)[12] intercept   : AIC=553.809, Time=4.94 sec
 ARIMA(0,1,2)(2,0,2)[12] intercept   : AIC=458.925, Time=3.70 sec
 ARIMA(0,1,2)(1,0,2)[12] intercept   : AIC=479.603, Time=2.96 sec
 ARIMA(0,1,2)(2,0,1)[12] intercept   : AIC=476.461, Time=2.77 sec
 ARIMA(0,1,2)(1,0,1)[12] intercept   : AIC=453.860, Time=1.27 sec
 ARIMA(0,1,2)(0,0,1)[12] intercept   : AIC=1590.168, Time=0.31 sec
 ARIMA(0,1,2)(1,0,0)[12] intercept   : AIC=830.178, Time=0.50 sec
 ARIMA(0,1,2)(0,0,0)[12] intercept   : AIC=1857.657, Time=0.10 sec
 ARIMA(0,1,2)(0,0,2)[12] intercept   : AIC=1416.319, Time=0.82 sec
 ARIMA(0,1,2)(2,0,0)[12] intercept   : AIC=645.836, Time=1.72 sec
 ARIMA(0,1,1)(1,0,1)[12] intercept   : AIC=455.591, Time=1.25 sec
 ARIMA(0,1,3)(1,0,1)[12] intercept   : AIC=482.851, Time=1.79 sec
 ARIMA(1,1,1)(1,0,1)[12] intercept   : AIC=490.039, Time=1.22 sec
 ARIMA(0,1,2)(1,0,1)[12]             : AIC=452.110, Time=0.68 sec
 ARIMA(0,1,2)(0,0,1)[12]             : AIC=1591.429, Time=0.17 sec
 ARIMA(0,1,2)(1,0,0)[12]             : AIC=828.557, Time=0.26 sec
 ARIMA(0,1,2)(2,0,1)[12]             : AIC=454.097, Time=2.08 sec
 ARIMA(0,1,2)(1,0,2)[12]             : AIC=454.097, Time=2.13 sec
 ARIMA(0,1,2)(0,0,0)[12]             : AIC=1859.631, Time=0.07 sec
 ARIMA(0,1,2)(0,0,2)[12]             : AIC=1417.084, Time=0.41 sec
 ARIMA(0,1,2)(2,0,0)[12]             : AIC=644.127, Time=0.59 sec
 ARIMA(0,1,2)(2,0,2)[12]             : AIC=inf, Time=2.08 sec
 ARIMA(0,1,1)(1,0,1)[12]             : AIC=453.786, Time=0.73 sec
 ARIMA(1,1,2)(1,0,1)[12]             : AIC=465.368, Time=1.63 sec
 ARIMA(0,1,3)(1,0,1)[12]             : AIC=450.959, Time=1.15 sec
 ARIMA(0,1,3)(0,0,1)[12]             : AIC=1558.880, Time=0.29 sec
 ARIMA(0,1,3)(1,0,0)[12]             : AIC=824.871, Time=0.31 sec
 ARIMA(0,1,3)(2,0,1)[12]             : AIC=452.929, Time=2.26 sec
 ARIMA(0,1,3)(1,0,2)[12]             : AIC=452.938, Time=3.26 sec
 ARIMA(0,1,3)(0,0,0)[12]             : AIC=1764.405, Time=0.11 sec
 ARIMA(0,1,3)(0,0,2)[12]             : AIC=1412.784, Time=0.62 sec
 ARIMA(0,1,3)(2,0,0)[12]             : AIC=642.758, Time=0.68 sec
 ARIMA(0,1,3)(2,0,2)[12]             : AIC=inf, Time=2.39 sec
 ARIMA(1,1,3)(1,0,1)[12]             : AIC=459.327, Time=1.92 sec
 ARIMA(0,1,4)(1,0,1)[12]             : AIC=452.923, Time=1.21 sec
 ARIMA(1,1,4)(1,0,1)[12]             : AIC=454.961, Time=2.19 sec

Best model:  ARIMA(0,1,3)(1,0,1)[12]
Total fit time: 120.761 seconds
```

Out[9]:

```
<function pmdarima.arima.arima.ARIMA.summary(self)>
```

In [12]:

```
train=avg.iloc[:-12]
test=avg.iloc[-12:]
test
```

Out[12]:

```
2021-04-01    419.05
2021-05-01    419.13
2021-06-01    418.94
2021-07-01    416.96
2021-08-01    414.47
2021-09-01    413.30
2021-10-01    413.93
2021-11-01    415.01
2021-12-01    416.71
2022-01-01    418.19
2022-02-01    419.28
2022-03-01    418.81
Freq: MS, Name: average, dtype: float64
```

In [14]:

```
model=SARIMAX(train, order=(0,1,3), seasonal_order=(1,0,1,12))
results=model.fit()
```

In [16]:

```
results
```

Out[16]:

```
<statsmodels.tsa.statespace.sarimax.SARIMAXResultsWrapper at 0x2297fe92670>
```

In [17]:

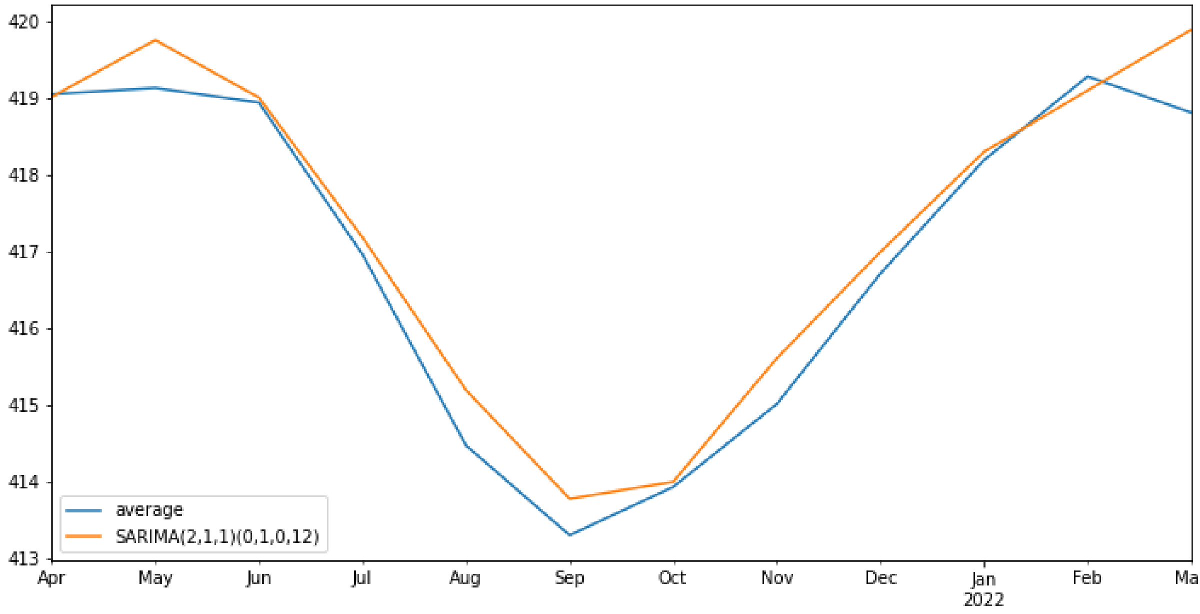
```
start=len(train)
end=start+len(test)-1
predictions=results.predict(start=start, end=end, dynamic=False).rename('SARIMA(2,1,
```

In [18]:

```
ax=test.plot(legend=True, figsize=(12,6))
predictions.plot(legend=True)
```

Out[18]:

```
<AxesSubplot:>
```



In [21]:

```
mean_absolute_percentage_error(test,predictions)
```

Out[21]:

```
0.0008913572782347107
```

In [ ]: