ts是index为日期的pd.Series

**AR**

<https://vitalflux.com/autoregressive-ar-models-with-python-examples/#:~:text=Autoregressive%20(AR)%20models%20are%20a,order%20to%20make%20accurate%20predictions>.

**VAR**

<https://www.machinelearningplus.com/time-series/vector-autoregression-examples-python/>

**滑动平均**

moving\_avg = ts.rolling(window).mean()

moving\_std= ts.rolling(window).std()

**ADF-test**

from statsmodels.tsa.stattools import adfuller

res=adfuller(ts)

res前四项：'Test Statistics', 'p-value', 'No. of lags used', 'Number of observations used'

res第五项：检验统计量bootstrap分布的1%,5%,10%分位值

**成分分解**

from statsmodels.tsa.seasonal import seasonal\_decompose

result = seasonal\_decompose(ts, model='additive', period=周期长度)

fig = plt.figure()

fig = result.plot()

**autoArima**

from pmdarima.arima import auto\_arima

model= auto\_arima(ts\_train, start\_p, start\_q, max\_p, max\_q,

test='adf', # use adftest to find optimal hyperparams

# maximum p and q

m=1, # frequency of series

d=None, # let model determine 'd'

seasonal=False, # No Seasonality

start\_P=0,

D=0,

trace=True, #Show selecting progress

error\_action='ignore',

suppress\_warnings=True,

stepwise=True)

print(model.summary())

model.plot\_diagnostics(figsize=(15,8))

plt.show()

**Arima**

from statsmodels.tsa.arima\_model import ARIMA

model = ARIMA(ts\_train, order=(p,d,q))

fitted = model.fit(disp=-1)

print(fitted.summary())

**预测**

pred, se, conf = fitted.forecast(预测步数, alpha=0.05)

pred\_series = pd.Series(pred, index=ts\_test.index)

lower\_series = pd.Series(conf[:, 0], index= ts\_test.index)

upper\_series = pd.Series(conf[:, 1], index= ts\_test.index)

**预测可视化**

plt.figure(figsize=(12,5), dpi=100)

plt.plot(ts\_train, label='training')

plt.plot(ts\_test, color = 'blue', label='Actual')

plt.plot(pred\_series, color = 'orange',label='Predicted')

plt.fill\_between(lower\_series.index, lower\_series, upper\_series, color='k', alpha=.10)

plt.legend(loc='upper left', fontsize=8)

plt.show()