

Never miss a tutorial:



Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python

[Click to Take the FREE Time Series Crash-Course](#)

How to Convert a Time Series to a Supervised Learning Problem in Python

Go from Data to Strategy

ONLINE MASTER OF SCIENCE IN BUSINESS ANALYTICS

Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Go from Data to Strategy: Tepper School of Business

Carnegie Mellon University
Tepper School of Business

Autoregression Models for Time Series Forecasting in Python

by **Jason Brownlee** on [January 2, 2017](#) in **Time Series**



Time Series Forecasting as Supervised Learning

[Share](#) [Share](#)

Last Updated on September 7, 2021

Autoregression is a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step.

The [Time Series with Python](#) EBook is

where you'll find the **Really Good** stuff. It is a very simple idea that can result in accurate forecasts on a range of time series problems.

In this tutorial >> [SEE WHAT'S INSIDE](#) how to implement an autoregressive model for time series forecasting with Python.

After completing this tutorial, you will know:


- How to explore your time series data for autocorrelation.
- How to develop an autocorrelation model and use it to make predictions.
- How to use a developed autocorrelation model to make rolling predictions.

Kick-start your project with my new book [Time Series Forecasting With Python](#), including *step-by-step tutorials* and the *Python source code* files for all examples.






Let's get started.

• **Updated May/2017:** Fixed small typo in autoregression equation

• Updated Apr/2020: Changed AR to AutoReg due to API change.
 Never miss a tutorial:



Picked for you:

-  [How to Create an ARIMA Model for Time Series Forecasting in Python](#)
-  [How to Convert a Time Series to a Supervised Learning Problem in Python](#)
-  [11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)
-  [How To Backtest Machine Learning Models for Time Series Forecasting](#)
-  [Time Series Forecasting as Supervised Learning](#)

Autoregression Models for Time Series Forecasting With Python
 Photo by Umberto Salvagnin, some rights reserved.

Autoregression

Loving the Tutorials?

A regression model, such as linear regression, models an output value based on a linear combination of input values. The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

For example:

>> SEE WHAT'S INSIDE

```
1 yhat = b0 + b1*X1
```

Where yhat is the prediction, b0 and b1 are coefficients found by optimizing the model on training data, and X is an input value.

This technique can be used on time series where input variables are taken as observations at previous time steps, called lag variables.

For example, we can predict the value for the next time step (t+1) given the observations at the last two time steps (t-1 and t-2). As a regression model, this would look as follows:

```
1 X(t+1) = b0 + b1*X(t-1) + b2*X(t-2)
```

Because the regression model uses data from the same input variable at previous time steps, it is referred to as an autoregression (regression of self).

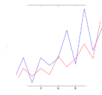
Never miss a tutorial! Stop learning Time Series Forecasting the *slow way*!



Take my free 7-day email course and discover how to get started (with sample code).

Click to sign-up and also get a free PDF Ebook version of the course.

Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python

Start Your FREE Mini-Course Now!



How to Convert a Time Series to a Supervised Learning Problem in Python



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Autocorrelation



How To Backtest Machine Learning Models for Time Series Forecasting

An autoregression model makes an assumption that the observations at previous time steps are useful to predict the value at the next time step.

This relationship between variables is called correlation.



Time Series Forecasting as Supervised Learning

When two variables change in the same direction (e.g. go up together or down together), this is called a positive correlation. If the variables move in opposite directions as values change (e.g. one goes up and one goes down), then this is called negative correlation.

We can use autocorrelation to calculate the correlation between the output variable and values at previous time steps at various different lags. The stronger the correlation between the output variable and a specific lagged variable, the more weight that autoregression model can put on that variable when modeling.

>> SEE WHAT'S INSIDE

Again, because the correlation is calculated between the variable and itself at previous time steps, it is called an autocorrelation. It is also called serial correlation because of the sequenced structure of time series data.

The correlation statistics can also help to choose which lag variables will be useful in a model and which will not.

Interestingly, if all lag variables show low or no correlation with the output variable, then it suggests that the time series problem may not be predictable. This can be very useful when getting started on a new dataset.

In this tutorial, we will investigate the autocorrelation of a univariate time series then develop an autoregression model and use it to make predictions.

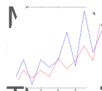
Never miss a tutorial:



Minimum Daily Temperatures Dataset

Picked for you:

This dataset describes the minimum daily temperatures over 10 years (1981-1990) in the city



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

The units are in degrees Celsius and there are 3,650 observations. The source of the data is credited as the Australian Bureau of Meteorology.



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Download the dataset into your current working directory with the filename “*daily-min-*



[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)

The code below will load the dataset as a Pandas Series.

```
1 from pandas import read_csv
2 from matplotlib import pyplot
3 series = read_csv('daily-min-temperatures.csv', header=0, index_col=0)
4 print(series.head())
5 series.plot()
6 pyplot.show()
```



[Time Series Forecasting as Supervised Learning](#)

ing the example prints the first 5 rows from the loaded dataset.

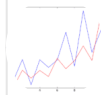
```
1 Date
2 1981-01-01 20.7
3 1981-01-02 17.9
4 1981-01-03 18.8
5 1981-01-04 14.6
6 1981-01-05 15.8
7 Name: Temp, dtype: float64
```

A line plot >> SEE WHAT'S INSIDE created.

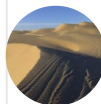
Never miss a tutorial:



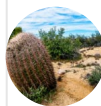
Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python



How to Convert a Time Series to a Supervised Learning Problem in Python



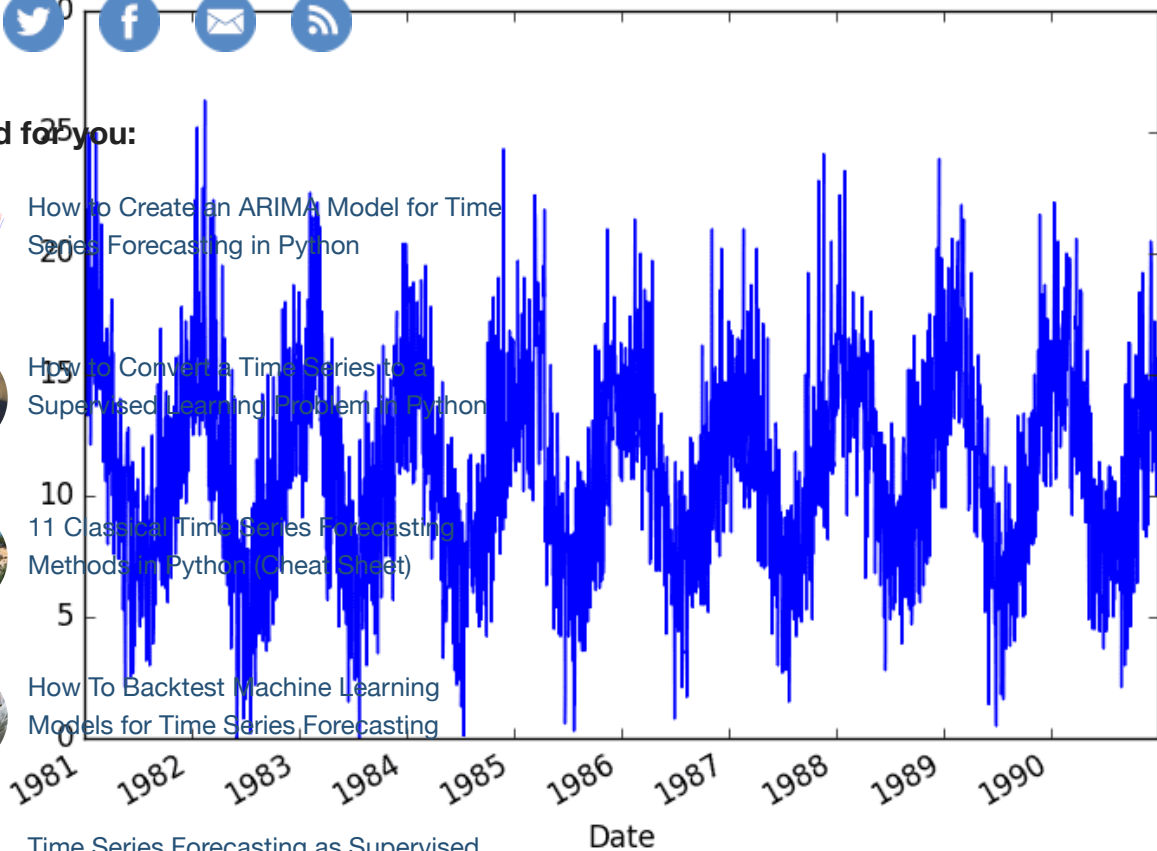
11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Learning



Minimum Daily Temperature Dataset Plot

Loving the Tutorials?

The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Quick Check for Autocorrelation

There is a quick, visual check that we can do to see if there is an autocorrelation in our time series dataset.

We can plot the observation at the previous time step ($t-1$) with the observation at the next time step ($t+1$) as a scatter plot.

This could be done manually by first creating a lag version of the time series dataset and using a built-in scatter plot function in the Pandas library.

But there is an easier way.

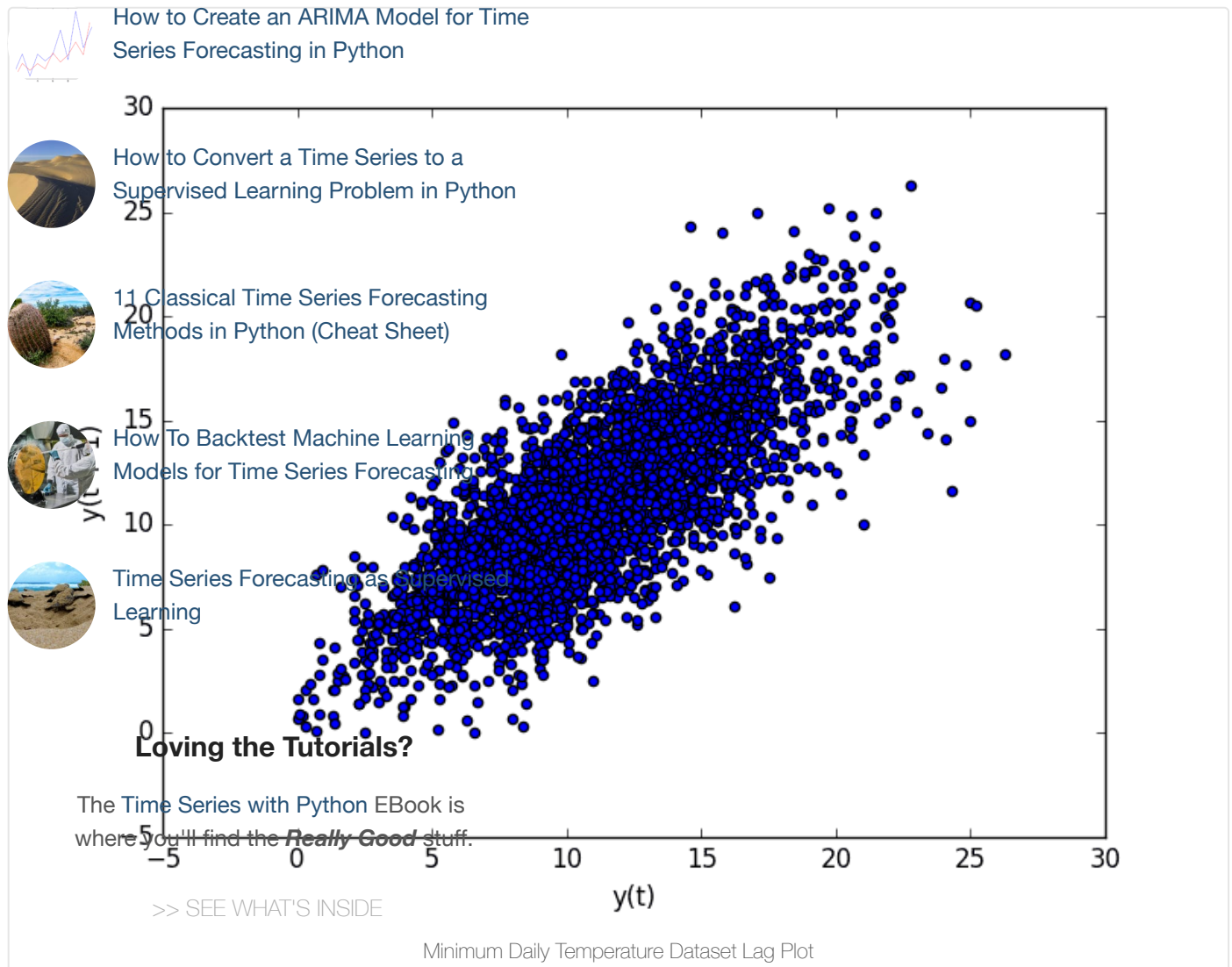
Pandas provides a built-in plot to do exactly this, called the `lag_plot()` function

```

2 from matplotlib import pyplot
3 from pandas import read_csv, lag_plot
4 series = read_csv('daily-min-temperatures.csv', header=0, index_col=0)
5 lag_plot(series)
6 pyplot.show()

```

Running the example plots the temperature data (t) on the x-axis against the temperature on the previous day ($t-1$) on the y-axis.



We can see a large ball of observations along a diagonal line of the plot. It clearly shows a relationship or some **correlation**.

This process could be repeated for any other lagged observation, such as if we wanted to review the relationship with the last 7 days or with the same day last month or last year.

Another quick check that we can do is to directly calculate the correlation between the observation and the lag variable.

We can use a statistical test like the **Pearson correlation coefficient**. This produces a number to summarize how correlated two variables are between -1 (negatively correlated) and +1 (positively correlated), with small values close to zero indicating low correlation and high values above 0.5 or

The example below creates a lagged version of the Minimum Daily Temperatures dataset and calculates a correlation matrix of each column with other columns, including itself.

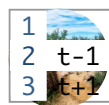


```
1 from pandas import read_csv
2 from pandas import DataFrame
3 from pandas import concat
4 from matplotlib import pyplot
5 series = read_csv('daily-min-temperatures.csv', header=0, index_col=0)
6 values = DataFrame(series.values)
7 dataframe = concat([values.shift(1), values], axis=1)
8 dataframe.columns = ['t-1', 't+1']
9 result = dataframe.corr()
10 print(result)
```



How to Convert a Time Series to a Supervised Learning Problem in Python

It shows a strong positive correlation (0.77) between the observation and the lag=1 value.



11 Classical Time Series Forecasting
t-1 1.00000 0.77487
t+1 0.77487 1.00000

This is good for one-off checks, but tedious if we want to check a large number of lag variables in our time series.



How To Backtest Machine Learning Models for Time Series Forecasting

Next, we will look at a scaled-up version of this approach.



Time Series Forecasting as Supervised Learning

Loving the Tutorials? Autocorrelation Plots

The Time Series with Python EBook is

where you will find the **Really Good** stuff.

This can v >> SEE WHAT'S INSIDE of which lag variables may be good candidates for use in a predictive model and how the relationship between the observation and its historic values changes over time.

We could manually calculate the correlation values for each lag variable and plot the result. Thankfully, Pandas provides a built-in plot called the `autocorrelation_plot()` function.

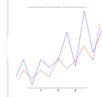
The plot provides the lag number along the x-axis and the correlation coefficient value between -1 and 1 on the y-axis. The plot also includes solid and dashed lines that indicate the 95% and 99% confidence interval for the correlation values. Correlation values above these lines are more significant than those below the line, providing a threshold or cutoff for selecting more relevant lag values.

```
1 from pandas import read_csv
2 from matplotlib import pyplot
3 from pandas.plotting import autocorrelation_plot
```

Running the example shows the swing in positive and negative correlation as the temperature values change across summer and winter seasons each previous year.



Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python



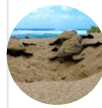
How to Convert a Time Series to a Supervised Learning Problem in Python



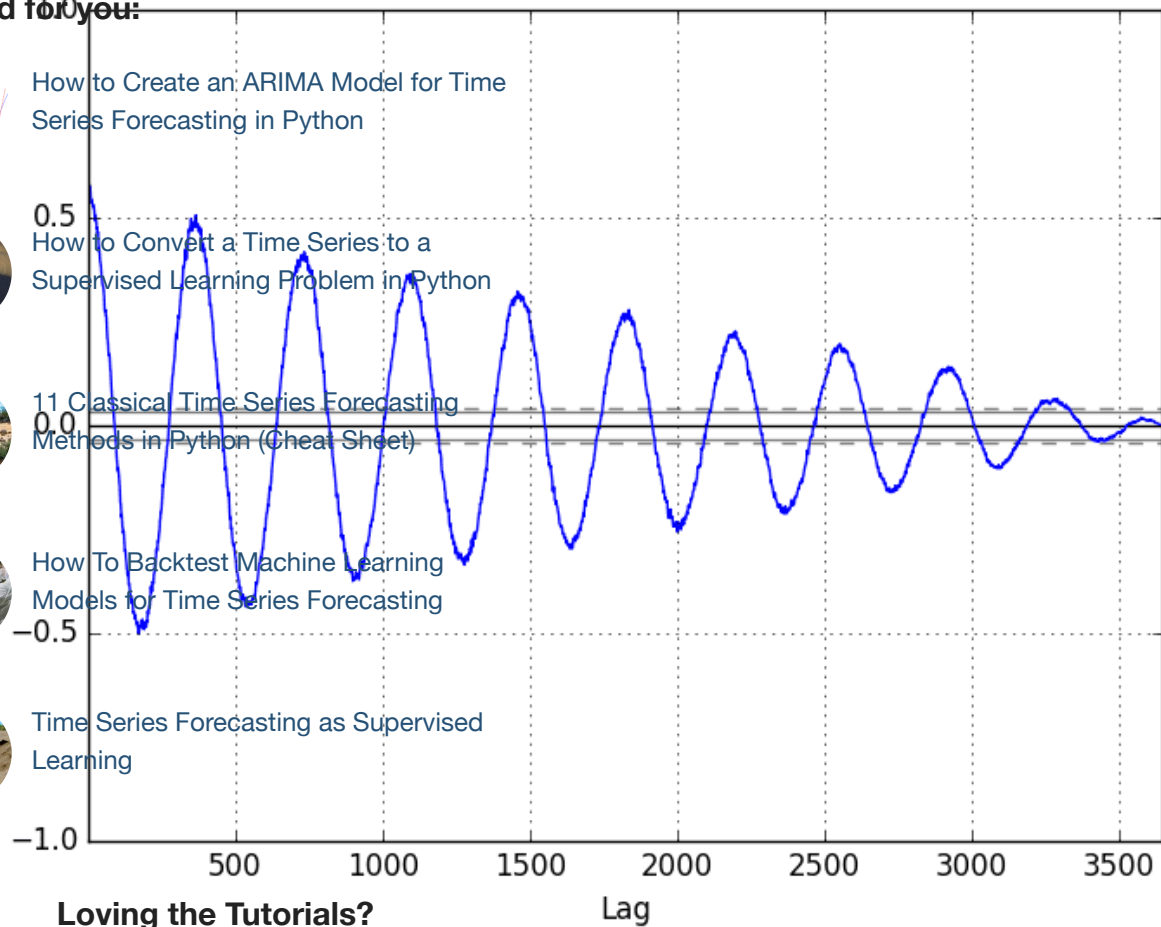
11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Learning



Loving the Tutorials?

The Time Series with Python EBook is where you'll find the **Really Good** stuff.

The statsmodels library also provides a version of the plot in the `plot_acf()` function as a line plot.

>> SEE WHAT'S INSIDE

```
1 from pandas import read_csv
2 from matplotlib import pyplot
3 from statsmodels.graphics.tsaplots import plot_acf
4 series = read_csv('daily-min-temperatures.csv', header=0, index_col=0)
5 plot_acf(series, lags=31)
6 pyplot.show()
```

In this example, we limit the lag variables evaluated to 31 for readability.

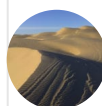
Never miss a tutorial:



Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python



How to Convert a Time Series to a Supervised Learning Problem in Python



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

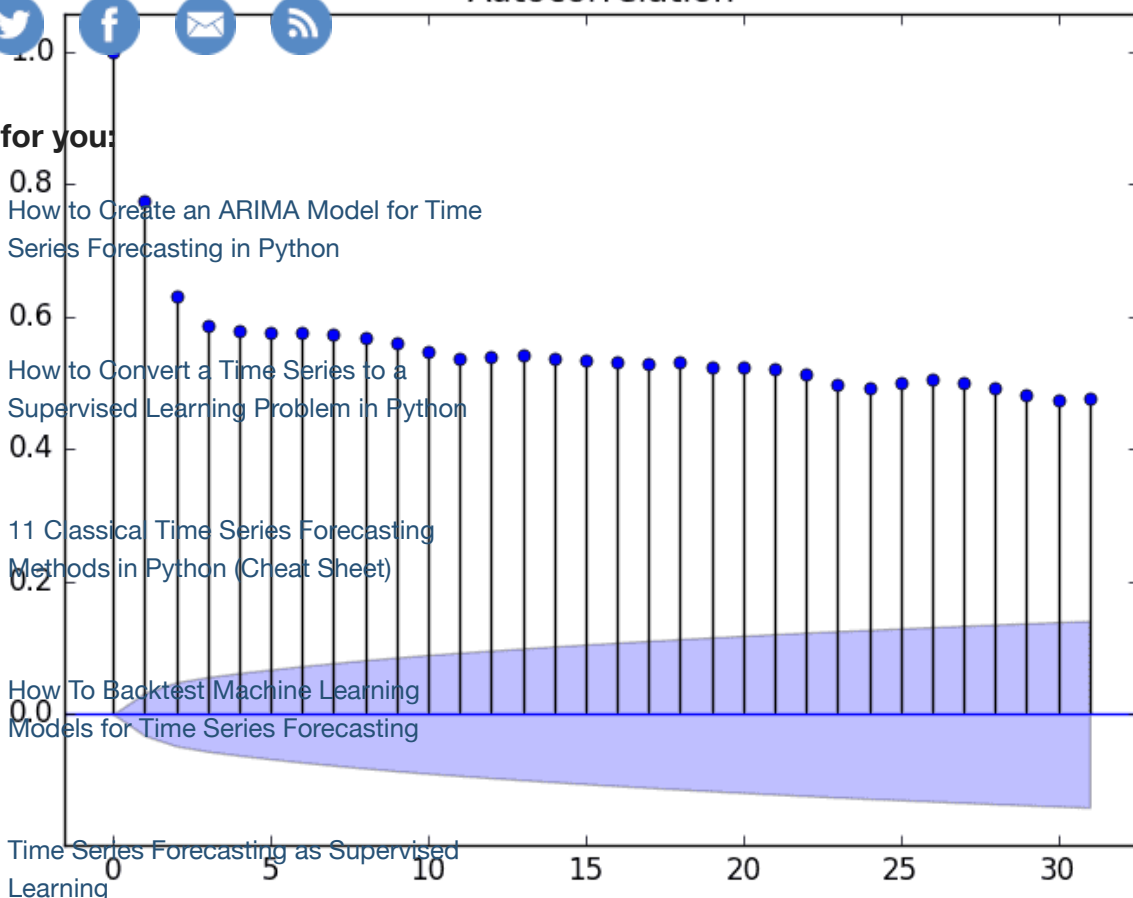


How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Learning

Autocorrelation



Statsmodels Autocorrelation Plot

Loving the Tutorials?

Now that we know how to review the autocorrelation in our time series, let's look at modeling it with an autoregression.

The Time Series with Python EBook is

where you'll find the **Really Good** stuff.

Before we do that, let's establish a baseline performance.

>> SEE WHAT'S INSIDE

Persistence Model

Let's say that we want to develop a model to predict the last 7 days of minimum temperatures in the dataset given all prior observations.

The simplest model that we could use to make predictions would be to persist the last observation. We can call this a persistence model and it provides a baseline of performance for the problem that we can use for comparison with an autoregression model.

The predictions are made using a walk-forward validation model so that we can persist the most recent observations for the next day. This means that we are not making a 7-day forecast, but 7 1-day



Never miss a tutorial:

```

1 from pandas import read_csv
2 from pandas import DataFrame
3 from pandas import concat
4 from matplotlib import pyplot
5 from sklearn.metrics import mean_squared_error
6 series = read_csv('daily-min-temperatures.csv', header=0, index_col=0)
7 # create lagged dataset
8 values = DataFrame(series.values)
9 dataframe = concat([values.shift(1), values], axis=1)
10 dataframe.columns = ['t-1', 't+1']
11 # split into train and test sets
12 X = dataframe.values
13 train, test = X[1:len(X)-7:], X[len(X)-7:]
14 train_X, train_y = train[:,0], train[:,1]
15 test_X, test_y = test[:,0], test[:,1]
16
17 # persistence model (Cheat Sheet)
18 def model_persistence(x):
19     return x
20
21 # walk-forward validation
22 predictions = list()
23 for x in test_X:
24     yhat = model_persistence(x)
25     predictions.append(yhat)
26 test_score = mean_squared_error(test_y, predictions)
27 print('Test MSE: %.3f' % test_score)
28 # plot predictions vs expected
29 pyplot.plot(test_y)
30 pyplot.plot(predictions, color='red')
31 pyplot.show()

```

Running the code prints the mean squared error (MSE).

The Time Series with Python EBook is where you'll find the **Really Good** stuff.

```
1 Test MSE: 3.423
```

>>> SEE WHAT'S INSIDE

The expected values for the next 7 days are plotted (blue) compared to the predictions from the model (red).

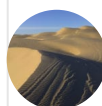
Never miss a tutorial:



Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python



How to Convert a Time Series to a Supervised Learning Problem in Python



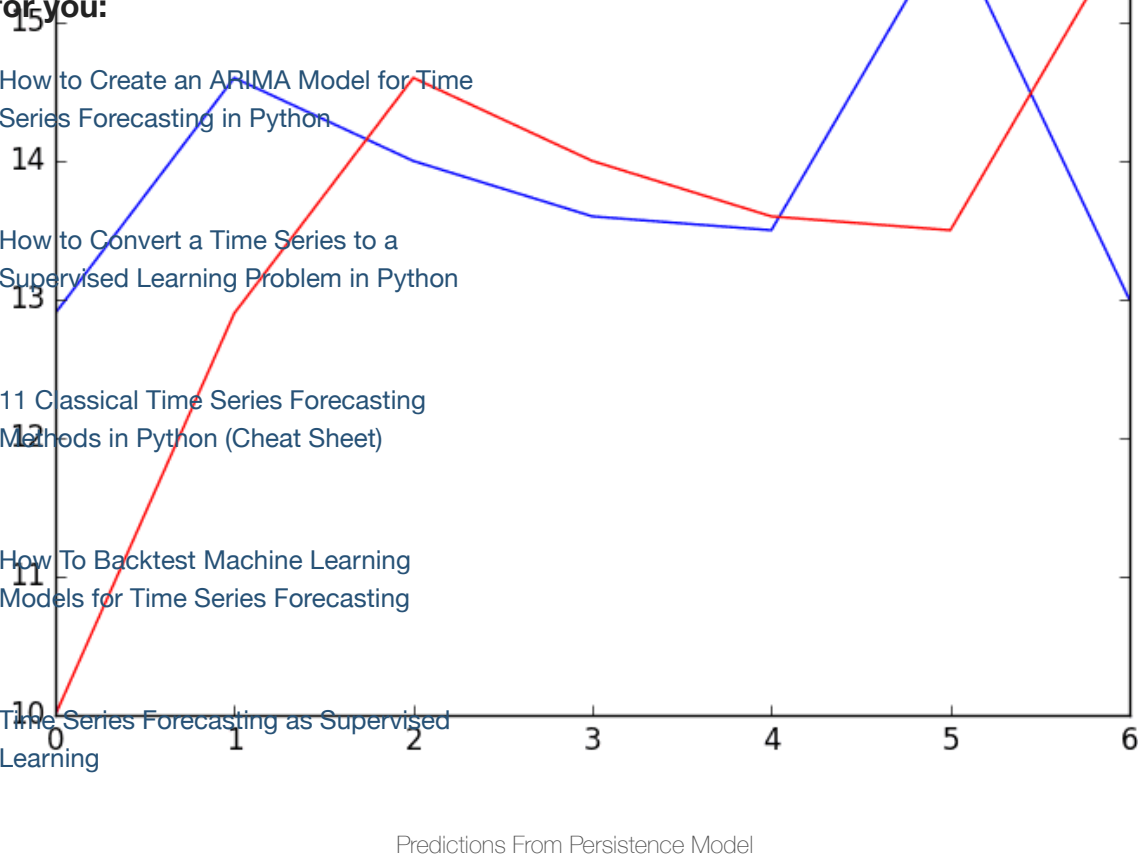
11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Learning



Predictions From Persistence Model

Loving the Tutorials?

The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Autoregression Model

An autoregression model is a linear regression model that uses lagged variables as input variables.

We could calculate the linear regression model manually using the `LinearRegression` class in `scikit-learn` and manually specify the lag input variables to use.

Alternately, the `statsmodels` library provides an autoregression model where you must specify an appropriate lag value and trains a linear regression model. It is provided in the `AutoReg` class.

We can use this model by first creating the model `AutoReg()` and then calling `fit()` to train it on our dataset. This returns an `AutoRegResults` object.

Once fit we can use the model to make a prediction by calling the `predict()` function for a number of

```

1 # create and evaluate a static autoregressive model
2 from pandas import read_csv
3 from matplotlib import pyplot
4 from statsmodels.tsa.ar_model import AutoReg
5 from sklearn.metrics import mean_squared_error
6 from math import sqrt
7 # load dataset
8 series = read_csv('daily-min-temperatures.csv', header=0, index_col=0, parse_dates=True, squeeze=True)
9 # split dataset
10 X = series.values
11 train, test = train_test_split(X, X[1:len(X)-7:])
12 # train autoregression
13 model = AutoReg(train, lags=29)
14 model_fit = model.fit()
15 print('Coefficients: %s' % model_fit.params)
16 # make predictions
17 predictions = model_fit.predict(start=len(train), end=len(train)+len(test)-1, dynamic=False)
18 for i in range(len(predictions)):
19     print('predicted=%f, expected=%f' % (predictions[i], test[i]))
20 rmse = sqrt(mean_squared_error(test, predictions))
21 print('Test RMSE: %.3f' % rmse)
22 # plot results
23 pyplot.plot(test)
24 pyplot.plot(predictions, color='red')
25 pyplot.show()

```



How To Backtest Machine Learning
Models for Time Series Forecasting

Using the example the list of coefficients in the trained linear regression model.

Models for Time Series Forecasting

The 7 day forecast is then printed and the mean squared error of the forecast is summarized.

```

1 Coefficients: [ 5.57543506e-01  5.88595221e-01 -9.08257090e-02  4.82615092e-02
2  4.00650265e-02  3.93020055e-02  2.59463738e-02  4.46675960e-02
3  1.27681498e-02  3.74362239e-02 -8.11700276e-04  4.79081949e-03
4  1.84731397e-02  2.68908418e-02  5.75906178e-04  2.48096415e-02
5  7.40316579e-03  9.91622149e-03  3.41599123e-02 -9.11961877e-03
6  2.42127561e-02  1.87870751e-02  1.21841870e-02 -1.85534575e-02
7 -1.77162867e-03  1.67319894e-02  1.97615668e-02  9.83245087e-03
8  6.22710723e-03 -1.37732255e-03]
9 predicted=11.871275, expected=12.900000
10 predicted=13.053794, expected=14.600000
11 predicted=13.532591, expected=14.000000
12 predicted=13.243126, expected=13.600000
13 predicted=13.091438, expected=13.500000
14 predicted=13.146989, expected=15.700000
15 predicted=13.176153, expected=13.000000
16 Test RMSE: 1.225

```

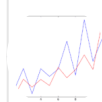
A plot of the expected (blue) vs the predicted values (red) is made.

The forecast does look pretty good (about 1 degree Celsius out each day), with big deviation on day 5.

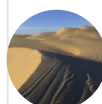
Never miss a tutorial:



Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python



How to Convert a Time Series to a Supervised Learning Problem in Python



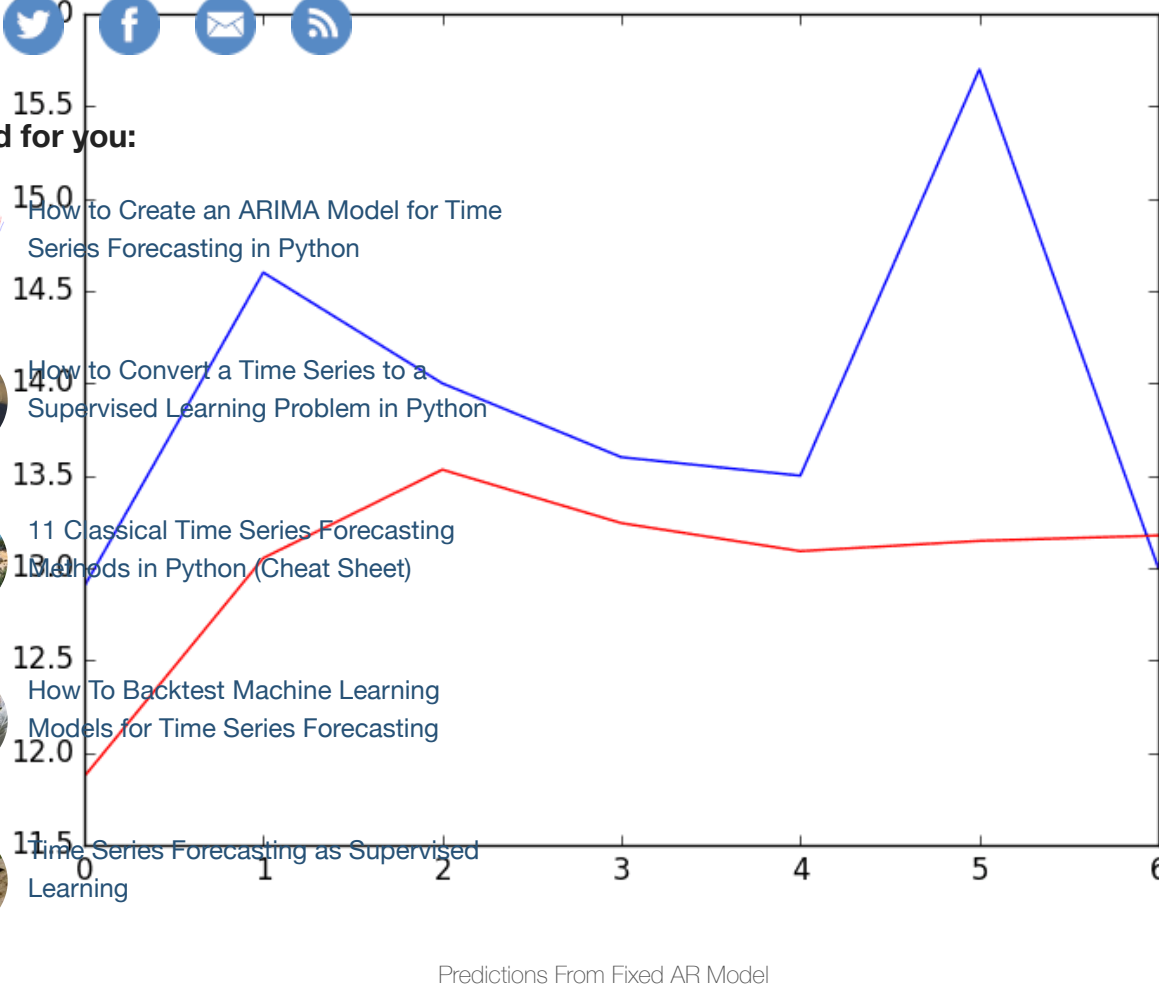
11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Learning



Loving the Tutorials?

The statsmodels API does not make it easy to update the model as new observations become available. The [Time Series with Python](#) EBook is

where you'll find the **Really Good** stuff.

One way would be to re-train the AutoReg model each day as new observations become available, and that may be >> SEE WHAT'S INSIDE it computationally expensive.

An alternative would be to use the learned coefficients and manually make predictions. This requires that the history of 29 prior observations be kept and that the coefficients be retrieved from the model and used in the regression equation to come up with new forecasts.

The coefficients are provided in an array with the intercept term followed by the coefficients for each lag variable starting at t-1 to t-n. We simply need to use them in the right order on the history of observations, as follows:

```
1 yhat = b0 + b1*X1 + b2*X2 ... bn*Xn
```

Below is the complete example.


```
1 # create and evaluate an updated autoregressive model
2 from pandas import read_csv
```

```

8 series = read_csv('daily-min-temperatures.csv', header=0, index_col=0, parse_dates=True, sq
9 # or miss a tutorial:
10 X = series.values
11 train, test = X[1:len(X)-7], X[len(X)-7:]
12 # train autoregression
13 window = 29
14 model = AutoReg(train, lags=29)
15 model_fit = model.fit()
16 coef = model_fit.params
17 # walk forward over time steps in test
18 history = train[len(train)-window:]
19 # history = history[-1:] for i in range(len(history))
20 predictions = list()
21 for t in range(len(test)):
22     length = len(history)
23     lag = [history[i] for i in range(length-window, length)]
24     yhat = coef[0]
25     for d in range(window):
26         yhat += coef[d+1] * lag[window-d-1]
27     obs = test[t]
28     predictions.append(yhat)
29     history.append(obs)
30     print('predicted=%f, expected=%f' % (yhat, obs))
31 rmse = sqrt(mean_squared_error(test, predictions))
32 print('Test RMSE: %.3f' % rmse)
33 # plot
34 pyplot.plot(test)
35 pyplot.plot(predictions, color='red')
36 pyplot.show()

```

Again, running the example prints the forecast and the mean squared error.

 Time Series Forecasting as Supervised Learning

```

1 predicted=12.871275, expected=12.900000
2 predicted=13.659297, expected=14.600000
3 predicted=14.349246, expected=14.000000
4 predicted=13.427454, expected=13.600000
5 predicted=13.374877, expected=13.500000
6 predicted=13.479991, expected=15.700000
7 predicted=14.765146, expected=13.000000
8 Test RMSE: 1.204

```

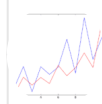
where you'll find the **Really Good** stuff.
We can see a small improvement in the forecast when comparing the error scores.

>> SEE WHAT'S INSIDE

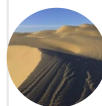
Never miss a tutorial:



Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python



How to Convert a Time Series to a Supervised Learning Problem in Python



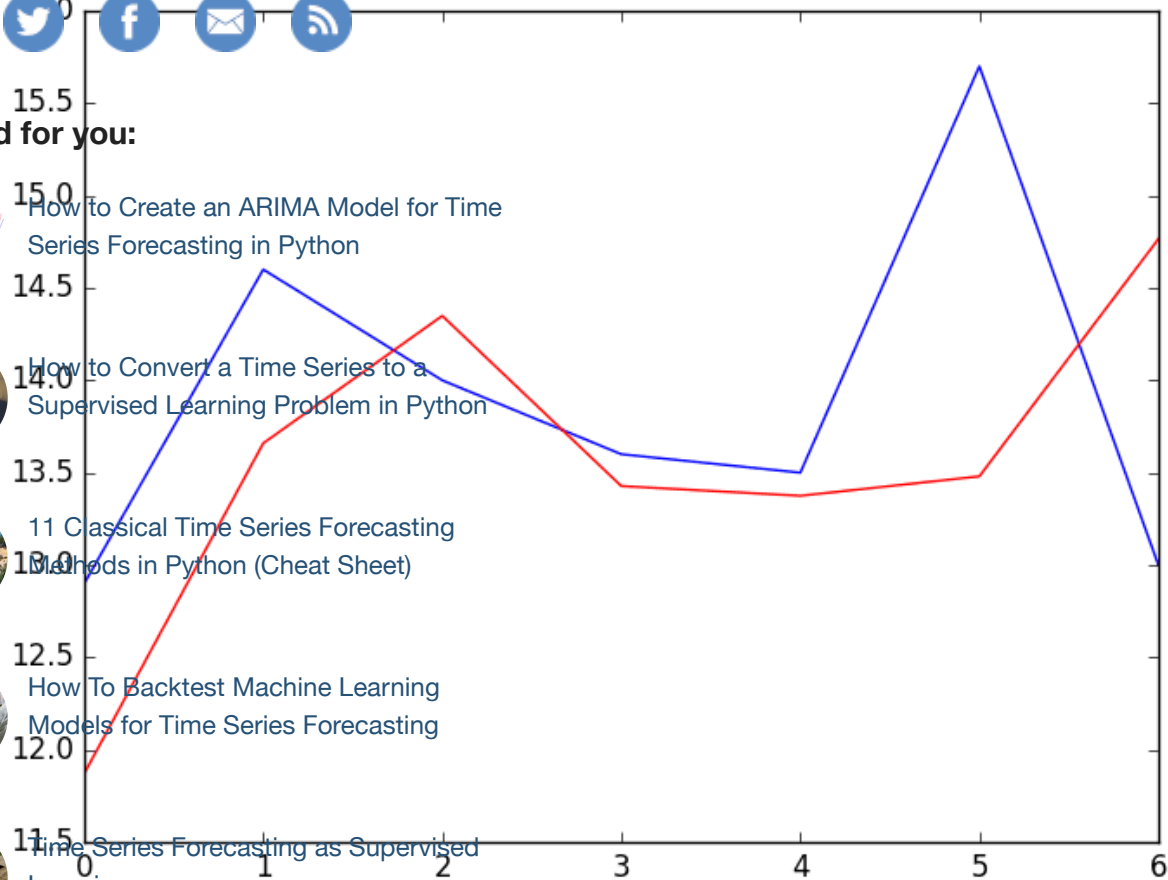
11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Learning



Predictions From Rolling AR Model

Loving the Tutorials?

The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Further Reading

This section provides some resources if you are looking to dig deeper into autocorrelation and autoregression.

- [Autocorrelation](#) on Wikipedia
- [Autoregressive model](#) on Wikipedia
- Chapter 7 – Regression-Based Models: Autocorrelation and External Information, [Practical Time Series Forecasting with R: A Hands-On Guide](#).
- Section 4.5 – Autoregressive Models, [Introductory Time Series with R](#).

Summary

• About autocorrelation and autoregression and how they can be used to better understand time series data.

Never miss a tutorial:

- How to explore the autocorrelation in a time series using plots and statistical tests.
- How to train an autoregression model in Python and use it to make short-term and rolling forecasts.

Picked for you:

Do you have any questions about autoregression, or about this tutorial?

Ask your questions in the comments below and I will do my best to answer.



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Want to Develop Time Series Forecasts with Python?



Develop Your Own Forecasts in Minutes

...with just a few lines of python code

Discover how in my new Ebook:

[Introduction to Time Series Forecasting With Python](#)

It covers **self-study tutorials** and **end-to-end projects** on topics like: *Loading data, visualization, modeling, algorithm tuning*, and much more...

Finally Bring Time Series Forecasting to Your Own Projects

Skip the Academics. Just Results.

[SEE WHAT'S INSIDE](#)

The Time Series with Python Ebook is where you'll find the **Really Good** stuff.

[Tweet](#)

>> [SEE WHAT'S INSIDE](#)

[Share](#)

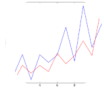
More On This Topic

Never miss a tutorial:



Picked for you:

How to Develop LSTM Models for Time Series Forecasting



How to Create an ARIMA Model for Time Series Forecasting in Python

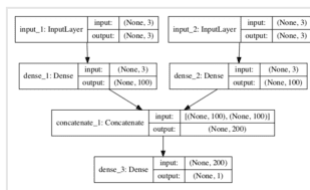


How to Convert a Time Series to a Supervised Learning Problem in Python

How to Develop Convolutional Neural Network Models...



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To Backtest Machine Learning Models for Time Series Forecasting

How to Develop Multilayer Perceptron Models for Time...



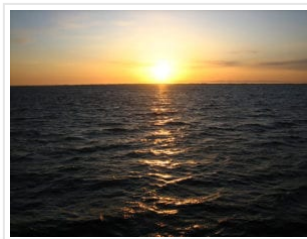
Time Series Forecasting as Supervised Learning



Loving the Tutorial?

The Time Series with Python EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE



How to Get Started with Deep Learning for Time...



How to Develop Multi-Step Time Series Forecasting...

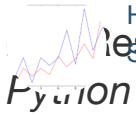
Never miss a tutorial:



Follow to keep in touch with the latest Data Science and Machine Learning tutorials with Python

Time Series Data Visualization with Python >

Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python

Python

Responses to Autoregression Models for Time Series Forecasting With



How to Convert a Time Series to a Supervised Machine Learning Problem in Python #

REPLY ↩

Thank you Jason for the awesome article



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)
If anyone hits the same problem I had – I downloaded the data from the link above as a csv file.

It was failing to be imported due to three rows in the temperature column containing ‘?’.

Once these were removed the data imported ok.



How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Machine Learning #

REPLY ↩

Thanks for the heads up Gary.

Loving the Tutorials?



yaser alsultan December 16, 2019 at 3:57 pm #

REPLY ↩

The Time Series with Python EBook is

where you'll find the **Really Good** stuff.
Hi man, what do you think the best model that can deal with hour

>> SEE WHAT'S INSIDE



Jason Brownlee December 17, 2019 at 6:29 am #

REPLY ↩

There are no best algorithms. I recommend testing a suite of methods in order to discover what works best for your specific dataset.



Tim Melino January 14, 2017 at 10:28 am #

REPLY ↩

Hey Jason, thanks for the article. How would you go about forecasting from the end of the file when expected value is not known?

Never miss a tutorial: Hi Tim, you can use `model.predict()` as in the example and specify the index of the time step to be predicted.



Picked for you: **Siddhant Bhambri** October 16, 2018 at 12:47 am #

REPLY ↩



How to Create an ARIMA Model for Time Series Forecasting in Python
 Hey, Jason, I am not clear with the use of `model.predict()`. If you could help me on this in predicting the values for the next 10 days if the model has learned the values till today.



How to Convert a Time Series to a Supervised Learning Problem
Jason Brownlee October 16, 2018 at 6:39 am #

REPLY ↩

Sure, see this post:

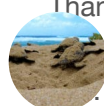


<https://machinelearningmastery.com/make-sample-forecasts-arma-python/>
 11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How to Build Machine Learning Models for Time Series Forecasting
 Hi Jason,

REPLY ↩



Thanks for all of your wonderful blogs. They are really helping a lot. One question regarding this post is I believe that AR modeling also presume that time series is stationary as the observations should be stationary. Does that AR function from statsmodels library checks for stationary and use the de-trended de-seasonalized time series by itself if required? Also, if we use scikit learn library for AR model as you described do we need to check for and make adjustments by ourselves for this?

Loving the Tutorials?

The **Time Series with Python** EBook is where you'll find the **Really Good** stuff.
Jason Brownlee January 26, 2017 at 4:45 am #

REPLY ↩

>> SEE WHAT'S INSIDE [the book](#).

The AR in statsmodels does assume that the data is stationary.

If your data is not stationary, you must make it stationary (e.g. differencing and other transforms).



Farrukh Jalali January 31, 2017 at 12:19 pm #

REPLY ↩

Thanks for the answer. Though we did not conduct proper test here for trend/seasonal stationarity check in the example above but from figure apparently it seems like that there is a seasonal effect. So in that case whether applying AR model is good to go?

Never miss a tutorial: AR is designed to be used on stationary data, meaning data with no seasonal or trend information.



Picked for you: **Farrukh Jalali** January 31, 2017 at 11:07 pm #

REPLY ↩



Or to be specific, is it OK to apply AR model direct here on the given data without checking the orality and removing it if present which is showing some signs in first graph apparently?

How to Convert a Time Series to a Supervised Learning Problem in Python
Muneeb November 11, 2020 at 11:19 am #

REPLY ↩

Hi. I am working on something similar and i have the same question? in the above AR model the normalised transformed time series has been used and it has not been made stationary? **Classical Time Series Forecasting Methods in Python (Chapters 8-9)**

How To Backtest Machine Learning Models for Time Series Forecasting
Jason Brownlee November 11, 2020 at 6:53 am #

REPLY ↩

It is a good idea to make the series stationary before using an AR model.

Time Series Forecasting as Supervised Learning

Anthony of Sydney March 8, 2017 at 5:09 am #

REPLY ↩

Loving the Tutorials?

I had a go at the 'persistence model' section.
 The Time Series with Python EBook is where you'll find the **Really Good** stuff.
 From your code

```
1 # persistence model
2 def model_persistence(x):
3     return x
4
5 # walk-forward validation
6 predictions = list()
7 for x in test_X:
8     yhat = model_persistence(x)
9     predictions.append(yhat)
10 test_score = mean_squared_error(test_y, predictions)
```

As soon as I try to compute the "test_score", I get the following error,

```
1 Traceback (most recent call last):
2   File "", line 1, in
3     test_score = mean_squared_error(test_y, predictions)
4   File "C:\Python34\lib\site-packages\sklearn\metrics\regression.py", line 232, in mean
5     output_errors = np.average((y_true - y_pred) ** 2, axis=0,
6   TypeError: ufunc 'subtract' did not contain a loop with signature matching types dtype(
```


Never miss a tutorial:**Jason Brownlee**

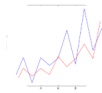
March 8, 2017 at 9:46 am #

REPLY ↩



books you need to convert your data to floating point values.

E.g. in numpy this might be:

Picked for you:`1 X = X.astype('float32')`[How to Create an ARIMA Model for Time Series Forecasting in Python](#)**Anthony of Sydney who noticed that it was a string not a float**

March 8, 2017 at 5:...

REPLY ↩

[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Dear Dr Jason,

I had the problem. You cannot assume that all *.csv numbers are floats or ints. For some reason, the numbers seem to be enclosed in quotes. Note that the data is for the sp500.csv not the above

exercise.

[11 Classical Time Series Forecasting](#)[Methods in Python \(Cheat Sheet\)](#)

the numbers in the output are enclosed in quotes:

```

1 test_y
2 array(['1.331', '1.412', '1.474', '1.559', '1.585', '1.788', '1.817'], dtype=object)
3
4 >>> predictions
5 ['1.24', '1.331', '1.412', '1.474', '1.559', '1.585', '1.788']
6 >>> test_y = [float(i) for i in test_y] # convert string to float
7 >>> predictions = [float(i) for i in predictions] # convert string to float
8 >>> test_score = mean_squared_error(test_y, predictions)
9 0.009805285714285716

```

[Time Series Forecasting as Supervised Learning](#)

works now,

Regards

Anthony from Sydney

Loving the Tutorials?[The Time Series with Python EBook](#) iswhere you'll find the **Really Good** stuff.**Jason Brownlee**

March 8, 2017 at 9:46 am #

REPLY ↩

>> SEE WHAT'S INSIDE

and it Anthony.

Anthony from Sydney

March 8, 2017 at 6:07 am #

REPLY ↩

Dear Dr Jason,

The problem has been fixed. The values in the array were strings, so I had to convert them to strings.

```

1 test_y
2 array(['1.331', '1.412', '1.474', '1.559', '1.585', '1.788', '1.817'], dtype=object)
3 >>> predictions
4 ['1.24', '1.331', '1.412', '1.474', '1.559', '1.585', '1.788']

```

So I converted the strings in each array to float.

`1 >>> test v = [float(i) for i in test v]`

Hope that helps others trying to convert values to appropriate data types in order to do numerical calculations.

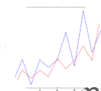
Never miss a tutorial:



Picked for you:

Johnson S April 17, 2017 at 6:39 am #

REPLY ↩



How to Create an ARIMA Model for Time

Series Forecasting in Python

How can I show my prediction as date, instead of log, for example I have data set incident number for each week, I want to predict the following week

week1 669



How to Convert a Time Series to a

Supervised Learning Problem in Python

so on april week 1 I want to show time series the prediction of week1 week2 of April



11 Classical Time Series Forecasting

Methods in Python (Cheat Sheet)

Jason Brownlee April 18, 2017 at 8:28 am #

REPLY ↩



How To Backtest Machine Learning

Models for Time Series Forecasting

Sorry, I'm not sure I understand. Perhaps you can give a fuller example of inputs and outputs?



Time Series Forecasting as Supervised

Learning

Chika April 30, 2017 at 2:46 am #

REPLY ↩

Thanks for this wonderful tutorial. I discovered your articles on facebook last year. Since then I have been following all your tutorials and I must confess that, though I started learning about machine learning in less than a year, my knowledge base has tremendously increased as a result of this free

Loving the Tutorials?

The **Time Series with Python** EBook is services you have been posting for all to see on the website.

where you'll find the **Really Good** stuff. Thanks once more for this generosity Dr Jason.

My que: >> SEE WHAT'S INSIDE Ily followed all your procedures in this article on my data set that was recorded every 10mins for 2 months. Please I would like to know which time lag is appropriate for forecasting to see the next 7 days value or more or less.

My time series is a stationary one according to the test statistics and histogram I have applied on it. but I still don't know if a reasonable conclusion can be reached with a data set that was recorded for 2 months every 10mins daily.

Jason Brownlee April 30, 2017 at 5:33 am #

REPLY ↩

Well done on your progress!

This post gives you some ideas on how to select suitable q and p values (lag vars):

<https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>

Never miss a tutorial:

Soy May 14, 2017 at 4:35 am #

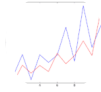
REPLY ↩



Dr Jason,

Thank you so much for this post. I have finally learned how to go from theory to practice.

Picked for you:



How to **Jason Brownlee** May 14, 2017 at 7:33 am #
Series Forecasting in Python

REPLY ↩

I'm so glad to hear that Soy, thanks!



How to Convert a Time Series to a
Supervised Learning Problem in Python

Akshit Mantri June 8, 2017 at 10:35 pm #

REPLY ↩



Dr. Jason, How do i predict the low and high confidence interval of prediction of an AR model?
11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To **Jason Brownlee** June 10, 2017 at 6:25 am #
Models for Time Series Forecasting

REPLY ↩

This post will help:

[https://machinelearningmastery.com/time-series-forecast-uncertainty-using-confidence-intervals-](https://machinelearningmastery.com/time-series-forecast-uncertainty-using-confidence-intervals-python/)



python/ Series Forecasting as Supervised
Learning

Akshit Mantri June 9, 2017 at 5:40 pm #
Loving the Tutorials?

REPLY ↩

Thanks for the reply. It was a very good post. But, it was for ARIMA model. I am having
The Time Series with Python EBook is some problem with the ARIMA model I cannot use it. Is there such a confidence interval
where you'll find the **Really Good** stuff.
forecasting for AR model?

>> SEE WHAT'S INSIDE

Jason Brownlee June 10, 2017 at 8:18 am #

REPLY ↩

Yes, I believe the same approach applies. Use the forecast() function.

Daoud June 11, 2017 at 9:10 pm #

REPLY ↩

ValueError: On entry to DLASCL parameter number 4 had an illegal value

I ma found above error when i use

model = AR(train) ## no error

model_fit = model.fit() ## show above error

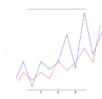
Never miss a tutorial:



Thanks for the great write-up Jason. One question though , I am interested in doing an autoregression on my timeseries data every nth days. For example. Picking the first 20 days

and predicting the value of the 20th day. Then picking the next 20 days (shift 1 over) and predicting the value of the 21st day and so forth until the end of my dataset. How can this be achieved in code? Thanks.

Picked for you:



How to Create an ARIMA Model for Time

Series Jason Brownlee June 15, 2017 at 8:54 am #

REPLY ↩



Consider using a variation of walk forward validation:

<https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/>
Supervised Learning Problem in Python



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet) June 19, 2017 at 9:49 pm #

REPLY ↩

Hi Jason,



Thanks for your article. I have few doubts.

How To Backtest Machine Learning

Models for Time Series Forecasting

1) We do analysis on the autocorrelation plots and auto-correlation function only after making the time series stationary right?



2) For the time series above, the correlation value is maximum for lag=1. So is it like the value at t-1 is more weight while doing Autoregression?

Time Series Forecasting as Supervised

Learning

3) When the AR model says, for lag-29 model the MSE is minimum. Does it mean the regression model constructed with values from t to t-29 gave minimum MSE?

Loving the Tutorials?

Please click

Thank You Time Series with Python EBook is

where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Jason Brownlee June 16, 2017 at 7:52 am #

REPLY ↩

1. Ideally, yes, analysis after the data is stationary.

2. Yes.

3. Yes.

Dhineshkumar June 16, 2017 at 6:46 pm #

REPLY ↩

Thanks.

Never miss a tutorial:



Tae Woo Kim June 21, 2017 at 10:10 am #

REPLY ↩

Hey Jason. Thanks for the awesome tutorial.

Picked for you:

in this line...



How to Create an ARIMA Model for Time Series Forecasting in Python

What is the “layman” explanation of what lag and coefficients are?



How to Convert a Time Series to a Supervised Learning Problem in Python

that “lag” is what the AR model determined that the significance ends (i.e. after this number, the correlation isn’t “strong enough”) and that the coeff. are the p-value of null hypothesis on “intensity” of the autocorrelation?



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Jason Brownlee June 21, 2017 at 8:08 am #

REPLY ↩



How To Backtest Machine Learning Models for Time Series Forecasting

Lag observations are observations at prior time steps.
Lag coefficients are the weightings on those lag observations.



Time Series Forecasting as Supervised Learning

TaeWoo Kim June 22, 2017 at 5:03 am #

REPLY ↩

Thank you Jason. That answer has still gotten me more confused. 😊

Loving the Tutorials?

The section where you manually calculate the predictions.. you’re specifying

The Time Series with Python EBook is where you’ll find the **Really Good** stuff.

In another words, this model can only predict 1 time period away ?

>> SEE WHAT’S INSIDE weak the code to predict up to n periods away?

TaeWoo Kim June 22, 2017 at 5:49 am #

REPLY ↩

i think i answered my own after reading this...

<https://machinelearningmastery.com/multi-step-time-series-forecasting/>

Example you gave with statsmodels.tsa.ar_model.AR is for single step prediction, am I correct?

Jason Brownlee June 22, 2017 at 6:13 am #

REPLY ↩

Never miss a tutorial! [Visit Yezinglu](#) September 30, 2017 at 12:29 am #

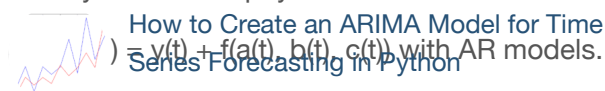
REPLY ↩



Thanks for this wonderful guideline. I have a problem with my dataset and I am digging in time series modeling.

Picked for you:

I try to model a physical model such that



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)
 $y(t) = f(a(t), b(t), c(t))$ with AR models.

Actually problem is predict metal temperature using basic heat transfer where y means metal

Temperature and f is a function of some sensors data like coolant mass flow rate, coolant temperaute, flowrate and gas temperature.



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Which model do you advise to use?



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Jason Brownlee September 30, 2017 at 7:43 am #

REPLY ↩



[How To Backtest Machine Learning Models for Time Series Forecasting](#)

Sounds like a great problem. I would recommend testing a suite of different methods to see what works best for your specific data, given your specific modeling requirements.

I recommend this process generally:



<https://machinelearningmastery.com/start-here/#process>

[Time Series Forecasting as Supervised Learning](#)

Monika October 13, 2017 at 3:48 am #

REPLY ↩

Loving the Tutorials?

Hi Jason...

I read [The Time Series with Python EBook](#) is helpful. I am very new to machine learning and would like to ask the meaning of these prediction points such as predicted=14.349246 so what is the meaning of this value does it mean???

>> SEE WHAT'S INSIDE

how does it help to understanding the prediction?

please post about cross regression also if you have posted any.

Jason Brownlee October 13, 2017 at 5:50 am #

REPLY ↩

Sorry, I don't follow. Perhaps you can rephrase your question?

Shantanu October 16, 2017 at 6:30 am #

REPLY ↩

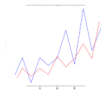
Thanks for the wonderful article.

ABC, PQR contribute in predicting prices. So I want to predict pricing based on these columns as well.
Never miss a tutorial:
 Thank you



Picked for you: **Jason Brownlee** October 16, 2017 at 3:46 pm #

REPLY ↩



It may be possible, but I do not have a worked example, sorry.
 How to Create an ARIMA Model for Time Series Forecasting in Python



Ryan Berger October 22, 2017 at 4:00 am #

REPLY ↩

How to Convert a Time Series to a Supervised Learning Problem in Python
 How could you make this a Deep Autoregressive Network with Keras?



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Jason Brownlee October 22, 2017 at 5:33 am #

REPLY ↩



A deep MLP will do the trick!
 How To Backtest Machine Learning Models for Time Series Forecasting



Foreign Forecasting October 23, 2017 at 3:28 am #
 Time Series Forecasting as Supervised Learning

REPLY ↩

Thank you for the amazing tutorial

I have a question though. According to Pandas' Autocorrelation plot, the maximum correlation is gained when lag=1. But the AR model selects lag=29 to build the autoregression.

I checked the autocorrelation plot, and the autoregression with lag=1 performed much better on test case than lag=14 chosen by AR model. Can you explain this? I thought that autocorrelation checks for linear relationship, thus, the autoregression which maps a linear function to the data should naturally perform best on the lag variable giving the maximum Pearson correlation.

The Time Series with Python EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Jason Brownlee November 10, 2017 at 10:37 am #

REPLY ↩

Perhaps the method was confused by noise in the data or small sample size?

Duplec November 19, 2017 at 12:45 am #

REPLY ↩

What if time is also included along with the date?

1981-01-01,20.7 is like 1981-01-01 03:00:00,20.7

Never miss a tutorial:



sam November 26, 2017 at 11:46 am #

REPLY ↩

Hi great tutorial. Just wanted to ask how would I change the order or lag in the code? Also if you had any tutorials for understanding how to use the statsmodels library.

Picked for you:



[How to Create an ARIMA Model for Time](#)

[Series Forecasting in Python](#)

Jason Brownlee

November 27, 2017 at 5:46 am #

REPLY ↩



My book is the best source of material on the topic:

<https://machinelearningmastery.com/introduction-to-time-series-forecasting-with-python/supervised-learning-problem-in-python/>

You can either difference your code directly or use the d parameter in the ARIMA model to control the differencing order. I have tutorials on both, perhaps start here:



[11 Classical Time Series Forecasting](#)

<https://machinelearningmastery.com/start-here/#timeseries>
[Methods in Python \(Cheat Sheet\)](#)



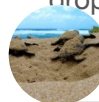
[How To Backtest Machine Learning](#)

[Models for Time Series Forecasting](#)

At December 31, 2017 at 4:25 am #

REPLY ↩

Thanks a lot! I ran into a problem while developing the AR model when some of the dates were dropped. I had to mention the frequency parameter even though I was already supplying the date-times



[Time Series Forecasting as Supervised](#)

[Learning](#)

Jason Brownlee

December 31, 2017 at 5:21 am #

REPLY ↩

Loving the Tutorials?

Interesting. Did it fix your issues?

The [Time Series with Python](#) EBook is

where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE 1:52 am #

REPLY ↩

Hi Jason,

Excellent article! Any chance of a blog post on how to do vector autoregression with big data?

Sometimes an LSTM is overkill, and even a vanilla RNN can be overkill, so something with just plain old autoregression would be great.

The VAR package in Python does this, but it runs into memory issues very quickly with large, sparse datasets.

If you know of any other tools for vector autoregression, any insight you have would be appreciated!

Jason Brownlee

March 14, 2018 at 6:31 am #

REPLY ↩

Harshil April 23, 2018 at 7:04 pm #

REPLY ↩

Never miss a tutorial:

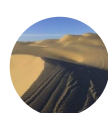


Dear Jason

Thank you so much for your wonderful article. I have a doubt regarding Data driven forecasting. I need to forecast appliance energy which depends upon 26 variables. I have data of appliance energy along with 26 variables of 3 months. With the help of 26 variables How can I forecast appliance energy for future?



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)



Jason Brownlee April 24, 2018 at 6:27 am #
How to Convert a Time Series to a Supervised Learning Problem in Python
Good question.

REPLY ↩

You can transform the data into a supervised learning problem and try a suite of machine learning



algorithms.

[10 Classical Time Series Forecasting Methods in Python \(Great Sheet\)](https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/)
<https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>

I hope to provide more information on this topic in the future.



[How To Backtest Machine Learning Models for Time Series Forecasting](#)

Praveen May 27, 2018 at 10:21 pm #

REPLY ↩



[Time Series Forecasting as Supervised Learning](#)
Hey Jason!

Thank you for the article. I have a doubt regarding this. How can I make predictions for future dates, that are not present in the dataset?

Loving the Tutorials?

The [Time Series with Python EBook](#) is where you'll find the *Really Good* stuff.

Jason Brownlee May 28, 2018 at 5:59 am #

REPLY ↩

Call `model.predict()` and specify the dates or index.

>> SEE WHAT'S INSIDE

This tutorial will show you how:

<https://machinelearningmastery.com/make-sample-forecasts-arma-python/>

Ashish June 21, 2018 at 5:43 pm #

REPLY ↩

can you please provide me a detaile example over VAR model?

Jason Brownlee June 22, 2018 at 6:03 am #

REPLY ↩

I hope to cover the method in the future, thanks for the suggestion.

Never miss a tutorial:
 from pandas import Series
 from matplotlib import pyplot



series = Series.from_csv('G:\Study_Material\daily-minimum-temperatures.csv')
 print(series.head())
 series.plot()
 pyplot.show()

Picked for you:



an error as : Empty 'DataFrame': no numeric data to plot
[How to Create an ARIMA Model for Time Series Forecasting in Python](#)
 to resolve it?



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Jason Brownlee

July 11, 2018 at 5:53 am #

REPLY ↩

Perhaps double check you have loaded the data correctly?



[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)

Vijay July 30, 2018 at 10:53 am #

[How To Backtest Machine Learning Models for Time Series Forecasting](#)

[Time Series Forecasting as Supervised Learning](#)

REPLY ↩



is a wonderful post that I share across and thanks a lot putting up great content with great examples. I am new to machine learning and I have a question regarding the use of ARIMA for sparse timeseries. I have events that can recur every day, week, once a few weeks, or monthly. Typical example is could be business meetings. Different meetings could happen at different frequencies. Is it appropriate to use ARIMA for predicting the underlying pattern. My real world problem involves predicting the size of virtual meetings based on their past history. Lets assume a service like hangout. I am trying to see if ARIMA would be an appropriate algorithm for predicting resource requirement for a virtual meeting based on its history. I tried ARIMA based on this tutorial but the results weren't convincing. I wasn't sure if its appropriate to model this problem as a time series problem and is ARIMA a good choice for this problem. The book is where you'll find the **Really Good** stuff.

Loving the Tutorials?

>> SEE WHAT'S INSIDE

Jason Brownlee July 30, 2018 at 2:16 pm #

REPLY ↩

Perhaps try it as a starting point.

Tim Boons August 7, 2018 at 6:53 pm #

REPLY ↩

I'm trying to work out o AR model to forecast a series using a lag of 192.

The series has a datapoint every 15 min but the you receive the data, the day after it was measured. So you have to forecast the next day (D+1) with data of the previous day (D-1) hence a lag of 192 in datapoints that are 15 min apart.

Is there a way to constrain the AR() function to all datapoints before t-192 ?

Perhaps fit a regularized linear regression model directly on your chosen lags?

Never miss a tutorial:

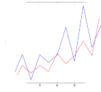


Tim Boons August 14, 2018 at 10:14 pm #

REPLY ↩

Picked for you:

Thanks for your reply!



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

Where can I find some more information about a regularized linear regression model ?



Jason Brownlee August 15, 2018 at 6:02 am #
[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

REPLY ↩

Any good book on machine learning, for example:

<https://amzn.to/2KSoQ0a>

[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)



Kamal Pokharel August 27, 2018 at 4:54 pm #
[How To Backtest Machine Learning Models for Time Series Forecasting](#)
Thanks a lot for this lovely article

REPLY ↩



Jason Brownlee August 28, 2018 at 5:57 am #
[Time Series Forecasting as Supervised Learning](#)

REPLY ↩

Thanks, I'm happy it helped.

Loving the Tutorials?

The [Time Series with Python](#) EBook is where you'll find the [Really Good](#) stuff. #

REPLY ↩

>> SEE WHAT'S INSIDE

Fantastic article — I'm following along step by step and it's helping.

at the step:

`model.fit()`

I get this error:

`TypeError: Cannot find a common data type.`

Where could this come from?

Jason Brownlee August 29, 2018 at 7:59 am #

REPLY ↩

That is odd. Are you using the code and data from the tutorial?

Never miss a tutorial:

Phil September 29, 2018 at 4:49 am #

REPLY ↩



Shouldn't your first equation be:

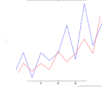
$$X(t) = b_0 + b_1 \cdot X(t-1) + b_2 \cdot X(t-2)$$

Picked for you:

Instead of:

$$X(t+1) = b_0 + b_1 \cdot X(t-1) + b_2 \cdot X(t-2)$$

[How to Create an ARIMA Model for Time Series Forecasting in Python](#)



Jason Brownlee September 29, 2018 at 6:38 am #
[How to Convert a Time Series to a Supervised Learning Problem in Python](#)
 Then where is "t"?

REPLY ↩



[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)
asieh August 28, 2019 at 6:30 am #

REPLY ↩



I have the same question as Phil, and I'm not sure what you mean by the answer. When you wrote: "we can predict the value for the next time step (t+1) given the observations at the last two time steps (t-1 and t-2)", the last two time steps for (t+1) are t, t-1 not t-1 and t-2. Why did you jump over t?



[Time Series Forecasting as Supervised Learning](#)

asieh August 28, 2019 at 6:35 am #

REPLY ↩

Loving the Tutorials?

also, in the following sentence you said it the right way:

"We can plot the observation at the previous time step (t-1) with the observation at the next time step (t+1) as a scatter plot."
 The [Time Series with Python EBook](#) is where you'll find the **Really Good** stuff.

So you are talking of current time (t) vs previous step (t-1) vs next step (t+1).
 >> SEE WHAT'S INSIDE
 THAT'S WHY I DON'T THINK the formula should be as Phil mentioned.

Jason Brownlee August 28, 2019 at 6:45 am #

Yes, thanks.

ML Uros October 12, 2018 at 2:12 am #

REPLY ↩

Hi Jason. Great job with your blog and this article!

I was wondering if you could help me with the following question: in your example, you choose 7 points

view on this matter. Does it mean that the AR model is not suitable for predictions too far in the future?

Never miss a tutorial:

Also, if I use the AR model for predicting about 180 points, AR's MSE value rises quite significantly, to about 9. Interestingly, if the test set is enlarged even more to about 350 points, MSE value falls to about 1. Persistence model's MSE has lower variability. What does this changing MSE say about the data and applying AR to it?

Picked for you:



How to Create an ARIMA Model for Time

Series **Jason Brownlee** October 12, 2018 at 6:42 am #

REPLY ↩

The further you predict into the future, the worse the performance.



How to Convert a Time Series to a Supervised Learning Problem in Python

Dieu Do October 13, 2018 at 1:54 pm #

REPLY ↩



11 Classical Time Series Forecasting

Methods in Python (Download Sheet)

Thanks a lot for your excellent article.

I have a question related to predicted results.



How To Backtest Machine Learning Models for Time Series Forecasting

the predicted solution at i-th point is very close the expected solution at (i-1)-th point ?

In most of your article, I have seen this.

I don't understand why? Can you help me answer this question, please?



Time Series Forecasting as Supervised Learning

Thank you so much Dr Jason Brownlee.

Best,

Dieu Do

Loving the Tutorials?

The **Time Series with Python** EBook is where you'll find the **Really Good** stuff.

Jason Brownlee

October 14, 2018 at 6:02 am #

REPLY ↩

http://>> SEE WHAT'S INSIDE

question that I answer here:

<https://machinelearningmastery.com/faq/single-faq/why-is-my-forecasted-time-series-right-behind-the-actual-time-series>

Yingfei October 22, 2018 at 1:21 am #

REPLY ↩

Hi Jason, thanks for your sharing.

I am trying to use AR model to predict a complex-valued time series. I used a series as below and replace the temperature data in your example code:

```
series = Series([1, 1+1j, 2, 3, 4, 5, 8, 1+2j, 3, 5])
```

However, it reports an error message like this:

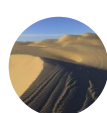
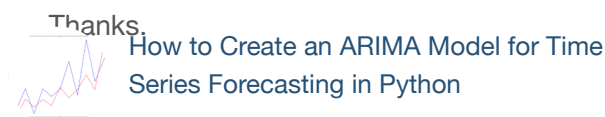
`ValueError: 0th/1st/2nd/3rd/4th/5th/6th/7th/8th/9th/10th/11th/12th/13th/14th/15th/16th/17th/18th/19th/20th/21st/22nd/23rd/24th/25th/26th/27th/28th/29th/30th/31st/32nd/33rd/34th/35th/36th/37th/38th/39th/40th/41st/42nd/43rd/44th/45th/46th/47th/48th/49th/50th/51st/52nd/53rd/54th/55th/56th/57th/58th/59th/60th/61st/62nd/63rd/64th/65th/66th/67th/68th/69th/70th/71st/72nd/73rd/74th/75th/76th/77th/78th/79th/80th/81st/82nd/83rd/84th/85th/86th/87th/88th/89th/90th/91st/92nd/93rd/94th/95th/96th/97th/98th/99th/100th/101st/102nd/103rd/104th/105th/106th/107th/108th/109th/110th/111st/112nd/113th/114th/115th/116th/117th/118th/119th/120th/121st/122nd/123rd/124th/125th/126th/127th/128th/129th/130th/131st/132nd/133th/134th/135th/136th/137th/138th/139th/140th/141st/142nd/143rd/144th/145th/146th/147th/148th/149th/150th/151st/152nd/153rd/154th/155th/156th/157th/158th/159th/160th/161st/162nd/163rd/164th/165th/166th/167th/168th/169th/170th/171st/172nd/173rd/174th/175th/176th/177th/178th/179th/180th/181st/182nd/183rd/184th/185th/186th/187th/188th/189th/190th/191st/192nd/193rd/194th/195th/196th/197th/198th/199th/200th/201st/202nd/203rd/204th/205th/206th/207th/208th/209th/210th/211st/212nd/213th/214th/215th/216th/217th/218th/219th/220th/221st/222nd/223rd/224th/225th/226th/227th/228th/229th/230th/231st/232nd/233th/234th/235th/236th/237th/238th/239th/240th/241st/242nd/243rd/244th/245th/246th/247th/248th/249th/250th/251st/252nd/253rd/254th/255th/256th/257th/258th/259th/260th/261st/262nd/263rd/264th/265th/266th/267th/268th/269th/270th/271st/272nd/273rd/274th/275th/276th/277th/278th/279th/280th/281st/282nd/283rd/284th/285th/286th/287th/288th/289th/290th/291st/292nd/293rd/294th/295th/296th/297th/298th/299th/300th/301st/302nd/303rd/304th/305th/306th/307th/308th/309th/310th/311st/312nd/313th/314th/315th/316th/317th/318th/319th/320th/321st/322nd/323rd/324th/325th/326th/327th/328th/329th/330th/331st/332nd/333th/334th/335th/336th/337th/338th/339th/340th/341st/342nd/343rd/344th/345th/346th/347th/348th/349th/350th/351st/352nd/353rd/354th/355th/356th/357th/358th/359th/360th/361st/362nd/363rd/364th/365th/366th/367th/368th/369th/370th/371st/372nd/373rd/374th/375th/376th/377th/378th/379th/380th/381st/382nd/383rd/384th/385th/386th/387th/388th/389th/390th/391st/392nd/393rd/394th/395th/396th/397th/398th/399th/400th/401st/402nd/403rd/404th/405th/406th/407th/408th/409th/410th/411st/412nd/413th/414th/415th/416th/417th/418th/419th/420th/421st/422nd/423rd/424th/425th/426th/427th/428th/429th/430th/431st/432nd/433th/434th/435th/436th/437th/438th/439th/440th/441st/442nd/443rd/444th/445th/446th/447th/448th/449th/450th/451st/452nd/453rd/454th/455th/456th/457th/458th/459th/460th/461st/462nd/463rd/464th/465th/466th/467th/468th/469th/470th/471st/472nd/473rd/474th/475th/476th/477th/478th/479th/480th/481st/482nd/483rd/484th/485th/486th/487th/488th/489th/490th/491st/492nd/493rd/494th/495th/496th/497th/498th/499th/500th/501st/502nd/503rd/504th/505th/506th/507th/508th/509th/510th/511st/512nd/513th/514th/515th/516th/517th/518th/519th/520th/521st/522nd/523rd/524th/525th/526th/527th/528th/529th/530th/531st/532nd/533th/534th/535th/536th/537th/538th/539th/540th/541st/542nd/543rd/544th/545th/546th/547th/548th/549th/550th/551st/552nd/553rd/554th/555th/556th/557th/558th/559th/560th/561st/562nd/563rd/564th/565th/566th/567th/568th/569th/570th/571st/572nd/573rd/574th/575th/576th/577th/578th/579th/580th/581st/582nd/583rd/584th/585th/586th/587th/588th/589th/590th/591st/592nd/593rd/594th/595th/596th/597th/598th/599th/600th/601st/602nd/603rd/604th/605th/606th/607th/608th/609th/610th/611st/612nd/613th/614th/615th/616th/617th/618th/619th/620th/621st/622nd/623rd/624th/625th/626th/627th/628th/629th/630th/631st/632nd/633th/634th/635th/636th/637th/638th/639th/640th/641st/642nd/643rd/644th/645th/646th/647th/648th/649th/650th/651st/652nd/653rd/654th/655th/656th/657th/658th/659th/660th/661st/662nd/663rd/664th/665th/666th/667th/668th/669th/670th/671st/672nd/673rd/674th/675th/676th/677th/678th/679th/680th/681st/682nd/683rd/684th/685th/686th/687th/688th/689th/690th/691st/692nd/693rd/694th/695th/696th/697th/698th/699th/700th/701st/702nd/703rd/704th/705th/706th/707th/708th/709th/710th/711st/712nd/713th/714th/715th/716th/717th/718th/719th/720th/721st/722nd/723rd/724th/725th/726th/727th/728th/729th/730th/731st/732nd/733th/734th/735th/736th/737th/738th/739th/740th/741st/742nd/743rd/744th/745th/746th/747th/748th/749th/750th/751st/752nd/753rd/754th/755th/756th/757th/758th/759th/760th/761st/762nd/763rd/764th/765th/766th/767th/768th/769th/770th/771st/772nd/773rd/774th/775th/776th/777th/778th/779th/780th/781st/782nd/783rd/784th/785th/786th/787th/788th/789th/790th/791st/792nd/793rd/794th/795th/796th/797th/798th/799th/800th/801st/802nd/803rd/804th/805th/806th/807th/808th/809th/810th/811st/812nd/813th/814th/815th/816th/817th/818th/819th/820th/821st/822nd/823rd/824th/825th/826th/827th/828th/829th/830th/831st/832nd/833th/834th/835th/836th/837th/838th/839th/840th/841st/842nd/843rd/844th/845th/846th/847th/848th/849th/850th/851st/852nd/853rd/854th/855th/856th/857th/858th/859th/860th/861st/862nd/863rd/864th/865th/866th/867th/868th/869th/870th/871st/872nd/873rd/874th/875th/876th/877th/878th/879th/880th/881st/882nd/883rd/884th/885th/886th/887th/888th/889th/890th/891st/892nd/893rd/894th/895th/896th/897th/898th/899th/900th/901st/902nd/903rd/904th/905th/906th/907th/908th/909th/910th/911st/912nd/913th/914th/915th/916th/917th/918th/919th/920th/921st/922nd/923rd/924th/925th/926th/927th/928th/929th/930th/931st/932nd/933th/934th/935th/936th/937th/938th/939th/940th/941st/942nd/943rd/944th/945th/946th/947th/948th/949th/950th/951st/952nd/953rd/954th/955th/956th/957th/958th/959th/960th/961st/962nd/963rd/964th/965th/966th/967th/968th/969th/970th/971st/972nd/973rd/974th/975th/976th/977th/978th/979th/980th/981st/982nd/983rd/984th/985th/986th/987th/988th/989th/990th/991st/992nd/993rd/994th/995th/996th/997th/998th/999th/1000th/1001st/1002nd/1003rd/1004th/1005th/1006th/1007th/1008th/1009th/1010th/1011st/1012nd/1013th/1014th/1015th/1016th/1017th/1018th/1019th/1020th/1021st/1022nd/1023rd/1024th/1025th/1026th/1027th/1028th/1029th/1030th/1031st/1032nd/1033th/1034th/1035th/1036th/1037th/1038th/1039th/1040th/1041st/1042nd/1043rd/1044th/1045th/1046th/1047th/1048th/1049th/1050th/1051st/1052nd/1053rd/1054th/1055th/1056th/1057th/1058th/1059th/1060th/1061st/1062nd/1063rd/1064th/1065th/1066th/1067th/1068th/1069th/1070th/1071st/1072nd/1073rd/1074th/1075th/1076th/1077th/1078th/1079th/1080th/1081st/1082nd/1083rd/1084th/1085th/1086th/1087th/1088th/1089th/1090th/1091st/1092nd/1093rd/1094th/1095th/1096th/1097th/1098th/1099th/1100th/1101st/1102nd/1103rd/1104th/1105th/1106th/1107th/1108th/1109th/1110th/1111st/1112nd/1113th/1114th/1115th/1116th/1117th/1118th/1119th/1120th/1121st/1122nd/1123rd/1124th/1125th/1126th/1127th/1128th/1129th/1130th/1131st/1132nd/1133th/1134th/1135th/1136th/1137th/1138th/1139th/1140th/1141st/1142nd/1143rd/1144th/1145th/1146th/1147th/1148th/1149th/1150th/1151st/1152nd/1153rd/1154th/1155th/1156th/1157th/1158th/1159th/1160th/1161st/1162nd/1163rd/1164th/1165th/1166th/1167th/1168th/1169th/1170th/1171st/1172nd/1173rd/1174th/1175th/1176th/1177th/1178th/1179th/1180th/1181st/1182nd/1183rd/1184th/1185th/1186th/1187th/1188th/1189th/1190th/1191st/1192nd/1193rd/1194th/1195th/1196th/1197th/1198th/1199th/1200th/1201st/1202nd/1203rd/1204th/1205th/1206th/1207th/1208th/1209th/1210th/1211st/1212nd/1213th/1214th/1215th/1216th/1217th/1218th/1219th/1220th/1221st/1222nd/1223rd/1224th/1225th/1226th/1227th/1228th/1229th/1230th/1231st/1232nd/1233th/1234th/1235th/1236th/1237th/1238th/1239th/1240th/1241st/1242nd/1243rd/1244th/1245th/1246th/1247th/1248th/1249th/1250th/1251st/1252nd/1253rd/1254th/1255th/1256th/1257th/1258th/1259th/1260th/1261st/1262nd/1263rd/1264th/1265th/1266th/1267th/1268th/1269th/1270th/1271st/1272nd/1273rd/1274th/1275th/1276th/1277th/1278th/1279th/1280th/1281st/1282nd/1283rd/1284th/1285th/1286th/1287th/1288th/1289th/1290th/1291st/1292nd/1293rd/1294th/1295th/1296th/1297th/1298th/1299th/1300th/1301st/1302nd/1303rd/1304th/1305th/1306th/1307th/1308th/1309th/1310th/1311st/1312nd/1313th/1314th/1315th/1316th/1317th/1318th/1319th/1320th/1321st/1322nd/1323rd/1324th/1325th/1326th/1327th/1328th/1329th/1330th/1331st/1332nd/1333th/1334th/1335th/1336th/1337th/1338th/1339th/1340th/1341st/1342nd/1343rd/1344th/1345th/1346th/1347th/1348th/1349th/1350th/1351st/1352nd/1353rd/1354th/1355th/1356th/1357th/1358th/1359th/1360th/1361st/1362nd/1363rd/1364th/1365th/1366th/1367th/1368th/1369th/1370th/1371st/1372nd/1373rd/1374th/1375th/1376th/1377th/1378th/1379th/1380th/1381st/1382nd/1383rd/1384th/1385th/1386th/1387th/1388th/1389th/1390th/1391st/1392nd/1393rd/1394th/1395th/1396th/1397th/1398th/1399th/1400th/1401st/1402nd/1403rd/1404th/1405th/1406th/1407th/1408th/1409th/1410th/1411st/1412nd/1413th/1414th/1415th/1416th/1417th/1418th/1419th/1420th/1421st/1422nd/1423rd/1424th/1425th/1426th/1427th/1428th/1429th/1430th/1431st/1432nd/1433th/1434th/1435th/1436th/1437th/1438th/1439th/1440th/1441st/1442nd/1443rd/1444th/1445th/1446th/1447th/1448th/1449th/1450th/1451st/1452nd/1453rd/1454th/1455th/1456th/1457th/1458th/1459th/1460th/1461st/1462nd/1463rd/1464th/1465th/1466th/1467th/1468th/1469th/1470th/1471st/1472nd/1473rd/1474th/1475th/1476th/1477th/1478th/1479th/1480th/1481st/1482nd/1483rd/1484th/1485th/1486th/1487th/1488th/1489th/1490th/1491st/1492nd/1493rd/1494th/1495th/1496th/1497th/1498th/1499th/1500th/1501st/1502nd/1503rd/1504th/1505th/1506th/1507th/1508th/1509th/1510th/1511st/1512nd/1513th/1514th/1515th/1516th/1517th/1518th/1519th/1520th/1521st/1522nd/1523rd/1524th/1525th/1526th/1527th/1528th/1529th/1530th/1531st/1532nd/1533th/1534th/1535th/1536th/1537th/1538th/1539th/1540th/1541st/1542nd/1543rd/1544th/1545th/1546th/1547th/1548th/1549th/1550th/1551st/1552nd/1553rd/1554th/1555th/1556th/1557th/1558th/1559th/1560th/1561st/1562nd/1563rd/1564th/1565th/1566th/1567th/1568th/1569th/1570th/1571st/1572nd/1573rd/1574th/1575th/1576th/1577th/1578th/1579th/1580th/1581st/1582nd/1583rd/1584th/1585th/1586th/1587th/1588th/1589th/1590th/1591st/1592nd/1593rd/1594th/1595th/1596th/1597th/1598th/1599th/1600th/1601st/1602nd/1603rd/1604th/1605th/1606th/1607th/1608th/1609th/1610th/1611st/1612nd/1613th/1614th/1615th/1616th/1617th/1618th/1619th/1620th/1621st/1622nd/1623rd/1624th/1625th/1626th/1627th/1628th/1629th/1630th/1631st/1632nd/1633th/1634th/1635th/1636th/1637th/1638th/1639th/1640th/1641st/1642nd/1643rd/1644th/1645th/1646th/1647th/1648th/1649th/1650th/1651st/1652nd/1653rd/1654th/1655th/1656th/1657th/1658th/1659th/1660th/1661st/1662nd/1663rd/1664th/1665th/1666th/1667th/1668th/1669th/1670th/1671st/1672nd/1673rd/1674th/1675th/1676th/1677th/1678th/1679th/1680th/1681st/1682nd/1683rd/1684th/1685th/1686th/1687th/1688th/1689th/1690th/1691st/1692nd/1693rd/1694th/1695th/1696th/1697th/1698th/1699th/1700th/1701st/1702nd/1703rd/1704th/1705th/1706th/1707th/1708th/1709th/1710th/1711st/1712nd/1713th/1714th/1715th/1716th/1717th/1718th/1719th/1720th/1721st/1722nd/1723rd/1724th/1725th/1726th/1727th/1728th/1729th/1730th/1731st/1732nd/1733th/1734th/1735th/1736th/1737th/1738th/1739th/1740th/1741st/1742nd/1743rd/1744th/1745th/1746th/1747th/1748th/1749th/1750th/1751st/1752nd/1753rd/1754th/1755th/1756th/1757th/1758th/1759th/1760th/1761st/1762nd/1763rd/1764th/1765th/1766th/1767th/1768th/1769th/1770th/1771st/1772nd/1773rd/1774th/1775th/1776th/1777th/1778th/1779th/1780th/1781st/1782nd/1783rd/1784th/1785th/1786th/1787th/1788th/1789th/1790th/1791st/1792nd/1793rd/1794th/1795th/1796th/1797th/1798th/1799th/1800th/1801st/1802nd/1803rd/1804th/1805th/1806th/1807th/1808th/1809th/1810th/1811st/1812nd/1813th/1814th/1815th/1816th/1817th/1818th/1819th/1820th/1821st/1822nd/1823rd/1824th/1825th/1826th/1827th/1828th/1829th/1830th/1831st/1832nd/1833rd/1834th/1835th/1836th/1837th/1838th/1839th/1840th/1841st/1842nd/1843rd/1844th/1845th/1846th/1847th/1848th/1849th/1850th/1851st/1852nd/1853rd/1854th/1855th/1856th/1857th/1858th/1859th/1860th/1861st/1862nd/1863rd/18`

-> 379 raise ValueError("maxlag should be < nobs")
 380 lm = np.zeros((nobs + maxlag, nvar * (maxlag + 1)))

381 for i in range(0, i * (maxlag + 1)):
 ValueError: maxlag should be < nobs

Please shed a light on how to correct it.

Picked for you:



Jason Brownlee October 22, 2018 at 6:20 am #

REPLY ↩

How to Convert a Time Series to a Supervised Learning Problem in Python

The error suggests you may need to change the configuration of the model to suite your data.



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

NikosLamp November 22, 2018 at 9:38 am #

REPLY ↩



How To Backtest Machine Learning Models for Time Series Forecasting

Hello
 In the section "Quick Check for Autocorrelation", you shifted the data by one position back and you named the columns 't-1' and 't+1'. In the article 'https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/' in the section 'Pandas shift() Function' you have the code 'df['t-1'] = df['t'].shift(1)' that is shifted by one means 1 time difference(t-1, t). Can you explain which one is correct? What point have I missed?

Thanks

Loving the Tutorials?

The Time Series with Python EBook is

where you'll find the **Really Good** stuff.

Jason Brownlee November 22, 2018 at 2:10 pm #

REPLY ↩

>> SEE WHAT'S INSIDE ' and 't-1.

Yannick December 13, 2018 at 9:24 pm #

REPLY ↩

Hi Dr. Brownlee,

I have used your tutorial to make some predictions on a dataset which records every minute the number of used parking spaces for 2017. For the Persistence model I get a test MSE score of 12.7 and for the Autoregression model a test MSE score of 74. Could you tell me if it is good or bad? In the meantime, could you give me more details on how the MSE results work on your code does it run on the entire dataset?

Regards.

Good or bad is only knowable in comparison to the persistence model.

Never miss a tutorial:

You can answer this question yourself.



7 12 = bad.

Picked for you:



Awilawa December 20, 2018 at 10:18 am #

REPLY ↩

How to Create an ARIMA Model for Time Series Forecasting in Python

Hi Dr. Brownlee,

Is there any references and example code of NARX (Non-Linear AutoRegressive with eXogenous inputs)



I apologize if this is out of topic, probably you have experience about this

How to Convert a Time Series to a Supervised Learning Problem in Python

Regards,



11 Classical Time Series Forecasting

Methods in Python (Cheat Sheet)

Jason Brownlee

December 20, 2018 at 2:02 pm #

REPLY ↩



I'm not sure, sorry.

How To Backtest Machine Learning

Models for Time Series Forecasting



Markus

December 23, 2018 at 10:07 pm #

REPLY ↩

Time Series Forecasting as Supervised

Learning

When we say that a given model makes use of lag value of 3, which one of the followings is the given model equation:

$$X(t) = b_0 + b_1 * X(t-1) + b_2 * X(t-2) + b_3 * X(t-3)$$

$$X(t) = b_0 + b_3 * X(t-3)$$

Loving the Tutorials?

The Time Series with Python EBook is I assume it's the first one, but I'm not sure. where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Jason Brownlee

December 24, 2018 at 5:29 am #

REPLY ↩

It will be a linear function of three prior time steps to the step being predictions.

The specifics of the linear function will vary across algorithms.

Victor

February 25, 2019 at 1:45 am #

REPLY ↩

Hi Jason,

How can I change the order for the AR model using this code.

E.g. AR(1), AR(2) etc?

Create the AR model and provide an integer for the order.

Never miss a tutorial:

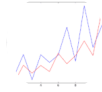


Victor February 25, 2019 at 9:12 am #

REPLY ↩

Picked for you:

I don't really understand what you mean?



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

I'd just like to know how to do it based on the example you gave. Wouldn't it just be a single line where I add the extra parameter as 'order number'?



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Victor February 25, 2019 at 1:28 pm #

REPLY ↩



[11 Classical Time Series Forecasting Methods in Python \(Cheatsheet\)](#)

would Python (Cheatsheet) based on the code above



[How To Backtest Machine Learning](#)

[Models for Time Series Forecasting](#) **Jason Brownlee** February 25, 2019 at 2:19 pm #

Yes, use maxlag on the fit() function or use an ARIMA without d or q elements.



[Time Series Forecasting as Supervised Learning](#)

Jason Brownlee February 25, 2019 at 2:15 pm #

REPLY ↩

Loving the Tutorials?

Just in case you set the "maxlag" argument on the call to fit(). More here: http://www.statsmodels.org/devel/generated/statsmodels.tsa.ar_model.AR.fit.html#statsmodels.tsa.ar_model.AR.fit The Time Series with Python EBook is where you'll find the **Really Good** stuff.

Alternately, you can use an ARIMA and set the order as (n, 0, 0).

>> SEE WHAT'S INSIDE

WiseNetAI March 6, 2019 at 6:02 am #

REPLY ↩

Could you please explain why it's not "cheating", so to speak, to append the observation at a time step to the history list. Isn't this equivalent to feeding the model part of the data it is trying to predict? Is this possible in real life situations where we may not know the actual value of the thing we are trying to predict? I am referring to line 26 in the code which generates the "Predictions From Rolling AR Model." I will appreciate it you could enlighten me about this please. Thank you.

Jason Brownlee March 6, 2019 at 8:00 am #

REPLY ↩

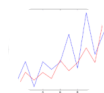
Never miss a tutorial: If this assumption does not hold for your data, you can design a walk forward validation strategy that captures the assumptions for your specific forecast problem.



Follow on social media for updates and notifications here:

<https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/>

Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python

WiseNetAI

March 7, 2019 at 11:32 am #

REPLY ↩

Thank you. That helped resolve my uncertainty.



How to Convert a Time Series to a Supervised Learning Problem in Python

Nihar Dharja

March 16, 2019 at 1:17 am #

REPLY ↩



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Hi,

model.predict(start, end) always gives me the same value and I end up getting a straight line prediction. So I used the history method shown above and kept adding yhat to test for the number of out of sample



How to Backtest Machine Learning Models for Time Series Forecasting

for i in range(len(test)+number_of_values_to_predict):

length = len(history)

lag = [history[i] for i in range(length-window,length)]



Time Series Forecasting as Supervised Learning

Learning

for d in range(window):

yhat += coef[d+1] * lag>window-d-1]

obs = test[t]

prediction.append(yhat)

history.append(obs)

The Time Series with Python EBook is

test = np.append(test,yhat)

where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Jason Brownlee

March 16, 2019 at 7:55 am #

REPLY ↩

I'm not really sure what you're trying to achieve?

Nihar Dharja

March 16, 2019 at 11:01 am #

REPLY ↩

I am trying to achieve out of sample forecasting like forecasting the value for the next 7 days. Using predict() gives me the same predicted value and gives me a straight line prediction. Hence I tried to use the code above. Do you think it is correct to do that?

Never miss a tutorial:



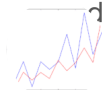
May 11 a Apr 17, 2019 at 1:38 pm #

REPLY ↩

Hey Jason,

Picked for you:

You are only providing an autoregression model in a case that you only have one label or one input which is previous records of the same output. What if for example we are concerned about the



prediction of energy consumption of a house and we have different input labels like indoor temp, outdoor temp and asking in Python. Consideration the home architecture and previous records.

How can we develop such a model?

do you have an example for that?



How to Convert a Time Series to a Supervised Learning Problem in Python



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet) April 5, 2019 at 6:13 am #

REPLY ↩

Good question, I refer to this as a multivariate time series problem, and you can find

examples here:



How To Backtest Machine Learning Models for Time Series Forecasting https://machinelearningmastery.com/start-here/#deep_learning_time_series



Time Series Forecasting as Supervised Learning April 17, 2019 at 10:54 am #

REPLY ↩

Hey Jason, I'm following your tutorial using my own dataset. But I have some questions about the results. (1)what does the Lag , that is the value of `model_fit.k_ar`, mean for your dataset?(2)what is the meaning of the period of Figure "Pandas Autocorrelation Plot" ? Could you take a moment to tell me something about them? Thank you very much.

The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE April 17, 2019 at 2:45 pm #

REPLY ↩

Lag is a prior observation, perhaps this will help:

<https://machinelearningmastery.com/time-series-forecasting-supervised-learning/>

More about autocorrelation here:

<https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>

Haiyang Duan April 20, 2019 at 11:50 am #

REPLY ↩

Could you please explain the relation between the `Lag(model_fit.k_ar)` and the period of dataset?

Never miss a tutorial:



How to Create an ARIMA Model for Time Series Forecasting in Python April 20, 2019 at 12:31 pm #

REPLY ↩

or how can I get the period of the time series dataset except Fast Fourier Transform?

Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python

Jason Brownlee April 21, 2019 at 8:19 am #

REPLY ↩



I see, you could review a plot of the series.
How to Convert a Time Series to a Supervised Learning Problem in Python
You could use domain knowledge.

You could use a grid search on a simple polynomial function.



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

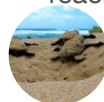
Dav April 24, 2019 at 8:39 pm #

REPLY ↩



Hi
How To Backtest Machine Learning Models for Time Series Forecasting

Thanks for this useful article. Where can I get the data file “daily-minimum-temperatures.csv”? I can’t reach the target site when I click on “Learn more about the dataset here”.



Time Series Forecasting as Supervised Learning

Loving the Tutorials?

Jason Brownlee April 25, 2019 at 8:11 am #

REPLY ↩

The Time Series with Python EBook is

You can download it directly from here:
where you'll find the **Really Good** stuff.

<https://raw.githubusercontent.com/jbrownlee/Datasets/master/daily-min-temperatures.csv>

>> SEE WHAT'S INSIDE

Dav May 1, 2019 at 3:15 am #

REPLY ↩

got it, thanks for your hlep

Thibault de Wit June 6, 2019 at 5:10 am #

REPLY ↩

Hi Jason,

Quick question here. Why do we start the train dataset at 1 and not 0?

In your code we have: train, test = X[1:len(X)-7], X[len(X)-7:]

But having this instead: train, test = X[:len(X)-7], X[len(X)-7:] would allow the model to train on 1 more

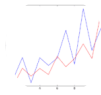
Never miss a tutorial:**Jason Brownlee**

June 6, 2019 at 6:39 am #

REPLY ↩



links. It's not sure what I was thinking there...

Picked for you:**Martin Muñoz Mandujano**

How to Create an ARIMA Model for Time Series Forecasting in Python July 2, 2019 at 12:41 pm #

REPLY ↩

Series Forecasting in Python

Hi Jason,

what order is the AR model in the code?

in what way is the order defined?



How to Convert a Time Series to a

Supervised Learning Problem in Python

**Jason Brownlee**

11 Classification Models for Time Series Forecasting July 2, 2019 at 2:10 pm #

REPLY ↩

Methods in Python (Cheat Sheet)

Order is the number of lag observations considered by the model.

You can grid search different values or use an ACF/PACF plot to choose the value.



How To Backtest Machine Learning

Models for Time Series Forecasting

**Ank**

July 28, 2019 at 7:07 pm #

REPLY ↩

Time Series Forecasting as Supervised

Learning

Hi Jason,

I am new to python, is it the update issue of pandas?

I cannot run the code for "Quick Check for Autocorrelation :"

Loving the Tutorials?

unless adding the below 2 line of code

The [Time Series with Python](#) EBook is
 Data['Date'] = pd.to_datetime(Data['Date'])
 where you'll find the **Really Good** stuff.
 series=pd.Series(Data['Temp'])

lag_plot >> SEE WHAT'S INSIDE

pyplot.show()

thank you

Jason Brownlee

July 29, 2019 at 6:12 am #

REPLY ↩

Sorry to hear that, what problem were you having exactly?

I have some suggestions here that might help:

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>

Thank you for the tutorial, very helpful. I have a question though

Never miss a tutorial:

How is your last example (rolling forecast) different to what `statsmodels.tsa.ar_model.AR.predict()` would



The way I understand it, it would do what you did in your last example, but you are getting different results, so not sure what I am missing.

Picked for you:

From statsmodels website:



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

If `dynamic` is `True`, then the in-sample lagged values are used for prediction. If `dynamic` is `False`, then the in-sample lagged dependent variables. The first forecasted value is `start`.”



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Jason Brownlee July 29, 2019 at 2:21 pm #

REPLY ↩



[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)

I believe `dynamic` only effects “in sample” data, e.g. data within the scope of the training data.



[How To Backtest Machine Learning Models for Time Series Forecasting](#)

Manpreet Singh September 9, 2019 at 8:49 am #

REPLY ↩



[Hi Jason Time Series Forecasting as Supervised Learning](#)

Thank you for the tutorial, very helpful. I have a question though

You wrote that “the statsmodels library provides an autoregression model that automatically selects an appropriate lag value using statistical tests and trains a linear regression model.”

Loving the Tutorials?

Is it using any model selection criteria like AIC, BIC to select the appropriate lag value or is it approximating all the models and picking the one with the least MSE? where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Manpreet Singh September 9, 2019 at 9:17 am #

REPLY ↩

```
fit([maxlag, method, ic, trend, ...])
```

Could you please explain what all inputs we're giving in this in your example. What is the `maxlag`, `method` and `ic` when we do `model.fit()`?

Jason Brownlee September 9, 2019 at 1:54 pm #

REPLY ↩

`maxlag` is the maximum input lag to consider when fitting the model.

The other parameters are described here:

http://www.statsmodels.org/stable/generated/statsmodels.tsa.ar_model.AR.fit.html#statsmodel

Never miss a tutorial:**Jason Brownlee** September 9, 2019 at 1:53 pm #

REPLY ↩



links, I glad helped.

Good question, you can see more about how it works here:

Picked for you:http://www.statsmodels.org/stable/generated/statsmodels.tsa.ar_model.AR.fit.html#statsmodels.tsa.ar_model.AR.fit

How to Create an ARIMA Model for Time Series Forecasting in Python

George September 11, 2019 at 4:43 am #

REPLY ↩

How to Convert a Time Series to a Supervised Learning Problem in Python
Hey Jason,

Do you know what the methods are for validating AR model?

Specific tests that could be done to show the model developed in valid

11 Classical Time Series Forecasting

Methods in Python (Cheat Sheet)

**Jason Brownlee** September 11, 2019 at 5:43 am #

REPLY ↩

How To Backtest Machine Learning Models for Time Series Forecasting
Yes, walk-forward validation:<https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/>

Time Series Forecasting as Supervised Learning

Leo September 24, 2019 at 2:56 pm #

REPLY ↩

Hi Jason,

Loving the Tutorials?

Thank you for the tutorial, very helpful. I have a question that your ACF lag is big in this case and so as I.

The Time Series with Python EBook is

How can we decide to choose the ARIMA parameters if the lag of ACF or PACF is very big ? In my case where you'll find the *Really Good* stuff.

, my ACF decay towards to zero at lag 1000, and PACF at lag 30.

>> SEE WHAT'S INSIDE

Jason Brownlee September 25, 2019 at 5:50 am #

REPLY ↩

Great question, I give some general advice for choosing ARIMA parameters from ACF/PACF plots here:

<https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>**Laguna** October 16, 2019 at 4:31 am #

REPLY ↩

Thank you for the article.

Isn't partial autocorrelation (PACF) plot supposed to be used to determine the statistically significant lag

Never miss a tutorial!

Jason Brownlee

October 16, 2019 at 8:12 am #


REPLY ↩



this tutorial

<https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>

Picked for you:




How to Create an ARIMA Model for Time Series Forecasting in Python

karan

February 10, 2020 at 9:01 pm #

REPLY ↩

Hi Jason. Very nice and clear explanation.



How to Convert a Time Series to a Supervised Learning Problem in Python

Jason Brownlee

February 11, 2020 at 5:08 am #

REPLY ↩




11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Thanks!

Evan Berry

March 2, 2020 at 8:00 am #

REPLY ↩




How To Build Machine Learning Models for Time Series Forecasting

Evan Berry

March 2, 2020 at 8:00 am #

REPLY ↩

Hi Jason, I was wondering if this technique of lag variables could be used for other features. So it seems like above you are predicting weather so you are using lag variables of weather data. What if I wanted to use lag variables of a supervised variable such as total sunlight per day. Even though this is a different variable can you use the say technique in terms of lagging data to predict weather?



Time Series Forecasting with Supervised Learning

Loving the Tutorials?

Jason Brownlee

March 2, 2020 at 10:07 am #

REPLY ↩

The Time Series with Python EBook is where you'll find the **Really Good** stuff. Yes, it can be useful for any time series data.

>> SEE WHAT'S INSIDE

Mark

March 28, 2020 at 1:18 am #

REPLY ↩

Hi Jason

I am trying to do a dynamic forecast using OLS. The model has an AR(1) variable and n exdog variables.

I use a for loop to run the regression many times for different combinations of the exdog variables and store the results. I split into test/train sets and run a prediction on the test set and store the MAE and RMSE.

The problem I have is the out of sample test is using the actual lagged AR(1) variable rather than dynamically generating it. I want to use the realized values of the exdog variables but the dynamically estimated AR(1) variable.

Do you have any thoughts on the best way of doing this please? I have looked at SARIMAX as an

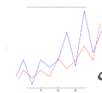
Cheers

Never miss a tutorial:

Mark


Picked for you: **Jason Brownlee** March 28, 2020 at 6:24 am #

REPLY ↩



How to Create an ARIMA Model for Time Series Forecasting in Python

I don't follow the specific problem you're having sorry. "generated" do you mean predicted as in a recursive model?

I cover the basics of time series forecasting with ARIMA in this book (no exog or sarima though):



<https://machinelearningmastery.com/introduction-to-time-series-forecasting-with-python/>

Supervised Learning Problem in Python

Not much on MC in python:

<https://machinelearningmastery.com/monte-carlo-sampling-for-probability/>



11 Classical Time Series Forecasting

Methods in Python (Cheat Sheet)

<https://machinelearningmastery.com/markov-chain-monte-carlo-for-probability/>



How To Backtest Machine Learning

Models for Time Series Forecasting

Mark March 28, 2020 at 9:16 am #

REPLY ↩

Thanks of the links Jason.



Time Series Forecasting as Supervised

Learning

I mean the standard way OLS works with statsmodels.predict() is to do a fixed forecast using the actual lagged dependent variables in the test data if there is a AR term in the equation. For a true out of sample forecast, for a given assumed exog data set or scenario, I want the forecast to be dynamic with regard to the AR term in the equation. Otherwise it looks better at out of

Loving the Tutorials? I really is.

The **Time Series with Python** EBook is

where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

vnlee March 29, 2020 at 5:48 am #

REPLY ↩

I don't follow, sorry. Perhaps you can rephrase your question/issue?

Vasiliki Voukelatou April 1, 2020 at 12:32 am #

REPLY ↩

Hi Mark amazing tutorial many thanks,

When using plot_acf(series, lags=30), I don't see why the autocorrelation plot appears 2 times. Is there a bug?

Thank you in advance for your reply.

Never miss a tutorial! Sorry to hear that you're having trouble, perhaps this will help:
<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me/>



Picked for you:

Vasiliki Voukelatou April 1, 2020 at 9:35 pm #

REPLY ↩



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

Hi Leon, who said Mark :P,

Thanks a lot, solved 😊



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Jason Brownlee April 2, 2020 at 5:50 am #

REPLY ↩



[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)

You did in your comment:

"Hi Mark amazing tutorial many thanks,"



[How To Backtest Machine Learning Models for Time Series Forecasting](#)

Happy to hear you solved your problem.



Arabzai April 17, 2020 at 7:29 pm #
[Time Series Forecasting as Supervised Learning](#)

REPLY ↩

Hi,

Can we fit Support vector regression instead of linear regression ?

If possible , then what is the procedure and how we can get the residuals from SVR-AR(1) model ?

Looking forward to hearing from you soon.

Loving the Tutorials?

Best,

Arabzai [The Time Series with Python](#) EBook is

where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE

Jason Brownlee April 18, 2020 at 5:45 am #

REPLY ↩

Yes, use this to prepare your data:

<https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>

Olly Price April 19, 2020 at 7:49 am #

REPLY ↩

Hi, do you know what statistical method `statsmodels.tsa.ar_model.AR()` uses under the hood to determine the optimal order for the AR? (Can't seem to find anything on the documentation.)

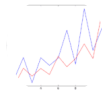
If not, are there any models you recommend me reading up on? I appreciate that you can observe an ACF and qualitatively decide a rough number.

Never miss a tutorial: The AR model does not optimize the order, you must specify the order when calling the fit() function.



Want a good idea of how to tune the model on your data, here is an example:
<https://machinelearningmastery.com/grid-search-arima-hyperparameters-with-python/>

Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python
 Only Price April 19, 2020 at 8:58 am #

REPLY ↩

If the AR model doesn't optimise the order, then where does model.fit.k_ar come from?



How to Convert a Time Series to a Supervised Learning Problem in Python
 model = AR(train)

model_fit = model.fit()

print("Lag: %s" % model_fit.k_ar)

11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)
 >>> Lag: 29

Where does the lag of 29 come from?



How To Backtest Machine Learning Models for Time Series Forecasting

Jason Brownlee April 19, 2020 at 9:05 am #

REPLY ↩



Time Series Forecasting as Supervised Learning
 looks like it tests different lag values using a configured criterion, learn more here:
https://www.statsmodels.org/stable/generated/statsmodels.tsa.ar_model.AR.fit.html#statsmodels.tsa.ar_model.AR.fit

Loving the Tutorials?

Robert Gadzinski May 17, 2020 at 2:25 am #
 The Time Series Analysis Python Book is where you'll find the **Really Good** stuff.

Thank you so much Jason, you saved my day

>> SEE WHAT'S INSIDE

REPLY ↩

Jason Brownlee May 17, 2020 at 6:37 am #

REPLY ↩

You're welcome!

Lucas June 7, 2020 at 7:26 pm #

REPLY ↩

Hello,

I am trying to follow the procedure, but I am having problems with statsmodels. All my dependencies are upto date, but I cannot import statsmodels, raising the error:

ImportError: cannot import name 'assert_equal' from 'statsmodels.compat pandas'

Never miss a tutorial:



Jason Brownlee June 8, 2020 at 6:09 am #

REPLY ↩

Perhaps confirm that pandas and statsmodels are up to date again?

Picked for you:



- 1 **pandas: 1.0.3** [Python ARIMA Model for Time Series Forecasting in Python](#)
- 2 **statsmodels: 0.11.1** [Series Forecasting in Python](#)



Ashwin June 25, 2020 at 11:28 pm #
How to Convert a Time Series to a Supervised Learning Problem in Python

REPLY ↩

I wanted to ask is AR(lags=10) model is equal to an ARMA(10,0)



Jason Brownlee June 26, 2020 at 5:36 am #
11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

REPLY ↩



Jason Brownlee June 26, 2020 at 5:36 am #
How To Backtest Machine Learning Models for Time Series Forecasting

REPLY ↩



Emy June 30, 2020 at 12:46 am #
Time Series Forecasting as Supervised Learning

REPLY ↩

hi how can i know numbre off the correct lag?

Loving the Tutorials?

The Time Series with Python EBook is where you'll find the **Really Good** stuff.

REPLY ↩

If you're unsure test a suite of values and use a number of lag obs that results in a model with >> SEE WHAT'S INSIDE

Cyril September 2, 2020 at 3:57 am #

REPLY ↩

Hi, Would you recommend using "statsmodels.tsa.ar_model.ar_select_order" to select best lag periods as "statsmodels.tsa.ar_model.AR" is now depreciated?

Jason Brownlee September 2, 2020 at 6:34 am #

REPLY ↩

I'm not familiar with statsmodels.tsa.ar_model.ar_select_order

AutoReg is an appropriate replacement:

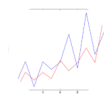
Never miss a tutorial: [krishnadas](#) July 5, 2020 at 3:16 am #

REPLY ↩



Nice example. This one helped me to start my VAR model project. Here, I'm using multivariate time series as a stats model's VAR model. As you mentioned the API won't update the coefficients for new observations. Since it's a multivariate time series, what can I do get the prediction for a long time (say 6 months)?

Picked for you:



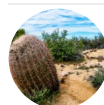
[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

Jason Brownlee July 5, 2020 at 7:07 am #

REPLY ↩



I believe you can call `forecast()` and predict as long as you like.
[How to Convert a Time Series to a Supervised Learning Problem in Python](#)
Then re-fit the model as you get new data.

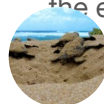


[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)
Dilip Thosar July 27, 2020 at 10:35 pm #

REPLY ↩



In most practical cases, we have some "regressable" variables in addition to the time series. Variables that are not uniformly periodic (seasonal) and generally in our control. eg. In a retail store sales forecasting application, "gift promotion scheme on (Y/N)" or "scheme discount percentage offered (% or \$)" may be significantly affecting the output variable, sales. Just running a timeseries model will ignore the effects of the schemes. How are these situations to be handled?



[Time Series Forecasting as Supervised Learning](#)

Jason Brownlee July 28, 2020 at 6:42 am #

REPLY ↩

Loving the Tutorials?

They can be included as exogenous variables to a linear model.
The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE [gust](#) 21, 2020 at 8:22 am #

REPLY ↩

When I run

```
from pandas import read_csv
from matplotlib import pyplot
from statsmodels.tsa.ar_model import AutoReg
from sklearn.metrics import mean_squared_error
from math import sqrt
# load dataset
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0, parse_dates=True,
squeeze=True)
# split dataset
X = series.values
train, test = X[1:len(X)-7], X[len(X)-7:]
```

Never miss a tutorial:

TypeError: Traceback (most recent call last)



```
11 train, test = X[1:len(X)-7], X[len(X)-7:]
```

```
12 # train autoregression
```

Picked for you:

```
> 13 model = AutoReg(train, lags=29)
```

```
14 model_fit = model.fit()
```

```
15 print('Coefficients: %s' % model_fit.params)
```

[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

```
16 conda3/lib/python3.7/site-packages/statsmodels/tsa/ar_model.py in __init__(self, endog, lags, trend, seasonal, exog, hold_back, period, missing)
```

```
17 self.hold_back = None, period=None, missing='none'):
```

[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

```
18 super(AutoReg, self).__init__(endog, exog, None, None, 165 missing=missing)
```

```
19 self._trend = string_like(trend, 'trend',
```

```
20 options=(('no', 'None'))
```

[Methods in Python \(Cheat Sheet\)](#)

```
21 conda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py in __init__(self, endog, exog, dates, freq, missing, **kwargs)
```

```
22 def __init__(self, endog, exog=None, dates=None, freq=None,
```

[How To Backtest Machine Learning Models for Time Series Forecasting](#)

```
23 missing='none', **kwargs):
```

```
24 super(TimeSeriesModel, self).__init__(endog, exog, missing=missing,
```

```
25 **kwargs)
```

[Time Series Forecasting as Supervised Learning](#)

```
26 Error: super(type, obj): obj must be an instance or subtype of type
```

Loving the Tutorials?

Jason Brownlee

August 21, 2020 at 8:23 am #

REPLY ↩

The [Time Series with Python](#) EBook is

where you can find the [Really Good](#) stuff.

This

[http](#)

me

>> SEE WHAT'S INSIDE

<https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me>

Giuseppe

October 14, 2020 at 1:24 am #

REPLY ↩

Hi Jason, thanks for the article.

I have a question, in the instruction `AutoReg(train, lags=29)` the lags parameter is equal to 29 because in the ACF graph we notice that there is a strong correlation up to 29 time?

Wouldn't it be better to consider the PACF graph?

Thank you for your answer.

Correct.

Never miss a tutorial:

See this on ACF/PACF plots:

<https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>**Picked for you:****Muhammad Zubair**

December 6, 2020 at 3:53 am #

REPLY ↩

[How to Create an ARIMA Model for Time](#)[Series Forecasting in Python](#)

Hi,

Nice explanation but I want to clarify that the time lags $t-1$ refers to one lag of time and the current time

are referring to is $t+1$. Then why don't you take it as t and $t+1$ or $t-1$ and t .

[How to Convert a Time Series to a](#)[Supervised Learning Problem in Python](#)

Now this is a very basic question but Appreciate your answer on it.

[11 Classical Time Series Forecasting](#)[Methods in Python \(Cheat Sheet\)](#)**Jason Brownlee**

December 6, 2020 at 7:10 am #

REPLY ↩

It is a good idea, I should do that.

[How To Backtest Machine Learning](#)[Models for Time Series Forecasting](#)**Girish**

March 21, 2021 at 1:16 am #

REPLY ↩

[Time Series Forecasting as Supervised](#)[Learning](#)

Hi Jason,

Thanks for very nice article. I am trying to perform the following , could you please suggest/guide me.

My original data is like below

Loving the Tutorials?

Month Price

The [Time Series with Python](#) EBook is

2011-01-01 1405.07

where you'll find the **Really Good** stuff.

2011-02-01 408.58

2011-03-01 277.29

2011-04-01 294.81

2011-05-01 511.77

2011-06-01 515.90

I have to predict the price (by regression) considering lag of 5, with the rebuilt the data set as below I need to predict the value for 5th element (X)

Dataset = [1405,408,277,294,X]

[408,277,294,511,X]

[277,294,511,515,X]

How I can achieve the same?

Thanks in advance

-Girish

Perhaps get started with time series forecasting generally here:
Never miss a tutorial!
<https://machinelearningmastery.com/start-here/#timeseries>



Picked for you: **Adarsh Krishna S** May 6, 2021 at 11:09 pm #

REPLY ↩



hi,
 How to Create an ARIMA Model for Time Series Forecasting in Python

I want a code for EEG signal prediction with autoregressive model , so what should be the changes in this temperature prediction .I am doing in jupyter notebook with MNE and Pandas.

Please help me.



How to Convert a Time Series to a Supervised Learning Problem in Python



Jason Brownlee May 7, 2021 at 6:27 am #
 11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

REPLY ↩

Perhaps try developing an ARIMA or SARIMA model for your dataset and compare the results to other methods.



How to Backtest Machine Learning Models for Time Series Forecasting
<https://machinelearningmastery.com/start-here/#timeseries>



Time Series Forecasting as Supervised Learning
Adarsh May 7, 2021 at 3:56 pm #

REPLY ↩

Sir it's my project. I want code for EAR model to do my project. So I was searching for AR model related to EEG and to modify AR to EAR but I am not getting code related to EEG
Loving the Tutorials?
 .can u help me.

The Time Series with Python EBook is where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE **vnlee** May 8, 2021 at 6:32 am #

REPLY ↩

Sorry, I cannot write code for you.

Perhaps you can adapt an example on the blog for your specific dataset.

Adarsh Krishna S May 10, 2021 at 4:47 pm #

REPLY ↩

sir,

I tried with EEG dataset with your autoregressive code and the output is fully different from expected and predicted values . Any solution?

Perhaps try an alternate model?

Never miss a tutorial:



Will Zhao May 23, 2021 at 7:52 am #

REPLY ↩

Picked for you:

Hello Jason,



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

You know how to modify the model set up so that we can put constraints on the coefficient matrix or example, in VAR(1), $Y_t = (A_1) (Y_{t-1}) + E$, $A_1 = [a_{11}, a_{12}; a_{21}, a_{22}]$, how to impose $a_{21} = 0$ since I

already know one granger cause the other, but not the other way around before running the model? I

also want granger causality test to reflect that too. Do you know the solution?

[How to Convert a Time Series to a](#)

[Supervised Learning Problem in Python](#)



11 Classical Time Series Forecasting

[Methods in Python \(Chart Sheet\)](#)

Jason Brownlee May 24, 2021 at 5:39 am #

REPLY ↩

Not off hand, sorry. It may require custom code.

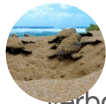


[How To Backtest Machine Learning](#)

[Models for Time Series Forecasting](#)

Shushman June 26, 2021 at 4:53 am #

REPLY ↩



[Time Series Forecasting as Supervised](#)

[Learning](#)

Thanks so much for the useful primer! I found lines 18-30 of the final code chunk a bit more verbose than necessary, in part because history ends up being an array of 1-length arrays rather than a flat array.

I rewrote that chunk as follows to get the same output:

Loving the Tutorials?

The [Time Series with Python EBook](#) is where you'll find the **Really Good** stuff.

```
history = train[len(train)-window:].flatten()
predictions = list()
for (t, obs) in enumerate(test.flatten()):
    hist_1 = history[-window:t+1]
    lag = np.concatenate(([1.0], np.flip(hist_1[1:window])))
    prediction = np.dot(coef, lag)
    predictions.append(prediction)
    history = np.append(history, obs)
print('predicted=%f, expected=%f' % (prediction, obs))
```

Jason Brownlee June 26, 2021 at 5:00 am #

REPLY ↩

You're welcome.

Thanks for sharing!

Hi Jason,

Never miss a tutorial:

Can we have scenario generations with this method? I mean several stochastic scenarios.

**Picked for you:**

Jason Brownlee [How to Model Time Series Forecasting in Python](#) August 10, 2021 at 5:25 am #

REPLY ↩

Yes, there are many examples on the blog, you can use the search box.



Jason Brownlee [How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Jason August 22, 2021 at 3:07 am #

REPLY ↩



Adrian Tam [11 Classical Time Series Forecasting Methods in Python \(Over Sheet\)](#) Thank you for this. Really good explanation of the AR model given that the article is 4yrs old. I have used Python (Over Sheet) using the ARIMA function on statsmodel, the lags and order the

same? Meaning you are using the autocorrelations of the past 29 time series to predict the next value?

So this comes out as an AR(29) model?



Adrian Tam [How To Backtest Machine Learning Models for Time Series Forecasting](#)

Adrian Tam August 23, 2021 at 5:10 am #

REPLY ↩



Adrian Tam [Time Series Forecasting as Supervised Learning](#)

you're referring to one of the example there, you're correct.

Loving the Tutorials?

Adrian Tam September 7, 2021 at 6:01 am #

REPLY ↩

The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

Please either correct the filename in the code pieces to read the CSV file

```
series = >>> SEE WHAT'S INSIDE -temperatures.csv', header=0, index_col=0)
```

or rename the CSV file now (currently: daily-min-temperatures.csv) to make filenames consistent and therefore avoiding Python error to not find the file.

Took me a little bit of time to realize why Python couldn't find the file after downloading and putting in the working directory.

Thanks.

Adrian Tam September 7, 2021 at 6:28 am #

REPLY ↩

Good catch. Thanks!

Could you tell a way to obtain the weights of this model.

Never miss a tutorial:



Adrian Tam November 14, 2021 at 1:22 pm #

REPLY ↩

Picked for you:

There are not weights in the model but there are coefficients. See the line:



`print('Coefficients: %s' % model.fit_params)`

[How to Create an ARIMA Model for Time Series Forecasting in Python](#)



Enesince December 29, 2021 at 8:53 am #

REPLY ↩

[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Hey! First of all Thank you for your articles and guides you really helped me a lot.

My question is, we constructed a model here and our forecast looks really good. But can you explain



[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)

that walk forward method works. I will write what I understand so please tell me if I'm right.

We fit our model and get coefficients first. And we have a set of train data with the size of window and it is at the end of train data which is called history. After that we started a loop for test set size and inside



[How To Backtest Machine Learning Models for Time Series Forecasting](#)

look at the test set and it is training set at first then we manually predict I don't get why didn't we use forecast() function but anyway we did that and we original value of predicted value to train set and continue so we keep predicting with only last 29 data and didn't change our model. I hope that's how it works.



[Time Series Forecasting as Supervised Learning](#)

I don't get is what's the point of that? We don't change our model we don't fit again what's the point? We keep using same coefficients. That just shows how well our model is trained and it only predicts 1 value everytime. Why don't we just go and predict upcoming 7 data instead doing this?

Loving the Tutorials?

The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

James Carmichael

January 20, 2022 at 8:50 am #

REPLY ↩

>> SEE WHAT'S INSIDE the following regarding your questions around walk forward validation:

<https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/>

Pablo March 13, 2022 at 6:09 am #

REPLY ↩

Hi just leaving my thanks for the article. Best wishes.

Pablo

James Carmichael March 13, 2022 at 1:09 pm #

REPLY ↩

Great feedback Pablo!

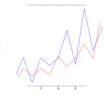
Never miss a tutorial:

One question.



In the persistence model, why do you define the function `model_persistence(x)`? When it does both out in the time series? You can plot `test_X` and `test_y` and it would be the same graph? Is it for notation/explanation only?

Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python

James Carmichael

March 13, 2022 at 1:20 pm #

REPLY ↩

Hi Pablo...You are correct. It serves just as an illustration.



How to Convert a Time Series to a Supervised Learning Problem in Python

Stellah

April 20, 2022 at 11:40 am #

REPLY ↩



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

Hello Jason, Many thanks again for the wonderful guideline. In the alternative where learned coefficients are used to manually make predictions. How can the code be modified to test out of sample

?

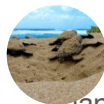


How To Backtest Machine Learning Models for Time Series Forecasting

Sophia

June 26, 2022 at 12:55 am #

REPLY ↩



Time Series Forecasting as Supervised Learning

Hi Jason,

Thank you for the article.

Snippet from your code:

```
model = AutoReg(train_lags=20)
model_fit = model.fit()
```

Loving the Tutorials?

The **Time Series with Python** EBook is where you'll find the **Really Good** stuff.

How did you determine lags = 29?

Is any way to let Statsmodels determine the best no for lags?

You might >> SEE WHAT'S INSIDE evaluate the model.

James Carmichael June 26, 2022 at 12:14 pm #

REPLY ↩

Hi Sophia...You may want to investigate hyperparameter optimization techniques.

<https://machinelearningmastery.com/combined-algorithm-selection-and-hyperparameter-optimization/>

GABZACHEW

September 7, 2022 at 12:27 am #

REPLY ↩

HI JASON

Never miss a tutorial:



Isadora November 9, 2022 at 11:44 am #

REPLY ↩

Hi James,

Picked for you: It was pretty helpfull.



if i understood correctly, what you call “rolling AR Model” is only rolling for the the validation on the set, but your training set is only one, right?

[How to Create an ARIMA Model for Time Series Forecasting in Python](#)

What do you think about fitting in a rolling-training-window? Do you have some code examples for that?



Thanks again!
[How to Convert a Time Series to a Supervised Learning Problem in Python](#)



11 Class James Carmichael November 10, 2022 at 7:34 am #

REPLY ↩

[Methods in Python \(Cheat Sheet\)](#)

Hi Isadora... You are very welcome! Your understanding of “rolling AR Model” is correct!



The following resource provides many code samples that will add clarity:

[How To Backtest Machine Learning Models for Time Series Forecasting](#)

<https://isadorafonseca.com/introduction-to-time-series-forecasting-with-python/>



Time Series Forecasting as Supervised Learning

Izzy January 30, 2023 at 5:48 am #

REPLY ↩

Hi there,

Is there a way to get the standard errors for the static and dynamic AR forecasts? Like in R the predict function for AR models has a ‘prediction’ element and a ‘standard error’ element so that you can plot the confidence intervals along with the prediction.

The [Time Series with Python EBook](#) is where you'll find the **Really Good** stuff.

This has been super helpful so far, thank you.

>> SEE WHAT'S INSIDE

James Carmichael January 30, 2023 at 9:38 am #

REPLY ↩

Hi Izzy...Some ideas in the following resource may be of interest to you:

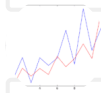
<https://www.geeksforgeeks.org/how-to-plot-a-confidence-interval-in-python/>

Leave a Reply

Never miss a tutorial:



Picked for you:



[How to Create an ARIMA Model for Time Series Forecasting in Python](#)



[How to Convert a Time Series to a Supervised Learning Problem in Python](#)

Name (required)

Email (will not be published) (required)



[11 Classical Time Series Forecasting Methods in Python \(Cheat Sheet\)](#)



[How To Backtest Machine Learning Models for Time Series Forecasting](#)

Welcome!

I'm Jason Brownlee PhD

and I **help developers** get results with **machine learning**.

[Read more](#)

[Time Series Forecasting as Supervised Learning](#)

Loving the Tutorials?

The [Time Series with Python](#) EBook is where you'll find the **Really Good** stuff.

>> [SEE WHAT'S INSIDE](#)

Never miss a tutorial:



Picked for you:



How to Create an ARIMA Model for Time Series Forecasting in Python



How to Convert a Time Series to a Supervised Learning Problem in Python



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)



How To Backtest Machine Learning Models for Time Series Forecasting



Time Series Forecasting as Supervised Learning

© 2023 Guiding Tech Media. All Rights Reserved.

[LinkedIn](#) | [Twitter](#) | [Facebook](#) | [Newsletter](#) | [RSS](#)

Loving the Tutorials?

[Privacy](#) | [Disclaimer](#) | [Terms](#) | [Contact](#) | [Sitemap](#) | [Search](#)

The Time Series with Python EBook is

where you'll find the **Really Good** stuff.

>> SEE WHAT'S INSIDE