How to Make Out-of-Sample Forecasts with ARIMA in Python

by **Jason Brownlee** on March 24, 2017 in **Time Series**

Last Updated on August 28, 2019

Making out-of-sample forecasts can be confusing when getting started with time series data.

The statsmodels Python API provides functions for performing one-step and multi-step out-of-sample forecasts.

In this tutorial, you will clear up any confusion you have about making out-of-sample forecasts with time series data in Python.

After completing this tutorial, you will know:

- How to make a one-step out-of-sample forecast.
- How to make a multi-step out-of-sample forecast.
- The difference between the *forecast()* and *predict()* functions.

Discover how to prepare and visualize time series data and develop autoregressive forecasting models in my new book, with 28 step-by-step tutorials, and full python code.

Let's get started.

- **Updated Apr/2019**: Updated the link to dataset.
- **Updated Aug/2019**: Updated data loading to use new API.

How to Make Out-of-Sample Forecasts with ARIMA in Python
Photo by dziambel, some rights reserved.

Tutorial Overview

This tutorial is broken down into the following 5 ste

1. Dataset Description
2. Split Dataset
3. Develop Model
4. One-Step Out-of-Sample Forecast
5. Multi-Step Out-of-Sample Forecast

1. Minimum Daily Temperatur

This dataset describes the minimum daily tempera
Melbourne, Australia.

The units are in degrees Celsius and there are 3,6 d
as the Australian Bureau of Meteorology.

- Download the dataset

Download the Minimum Daily Temperatures dataset to your current working directory with the filename "*daily-minimum-temperatures.csv*".

The example below loads the dataset as a Pandas Series.

```
# line plot of time series
from pandas import read_csv
from matplotlib import pyplot
# load dataset
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0)
# display first few rows
print(series.head(20))
# line plot of dataset
series.plot()
pyplot.show()
```

Running the example prints the first 20 rows of the loaded dataset.

```
Date
1981-01-01    20.7
1981-01-02    17.9
1981-01-03    18.8
1981-01-04    14.6
1981-01-05    15.8
1981-01-06    15.8
1981-01-07    15.8
1981-01-08    17.4
```

```
10 1981-01-09    21.8
11 1981-01-10    20.0
12 1981-01-11    16.2
13 1981-01-12    13.3
14 1981-01-13    16.7
15 1981-01-14    21.5
16 1981-01-15    25.0
17 1981-01-16    20.7
18 1981-01-17    20.6
19 1981-01-18    24.8
20 1981-01-19    17.7
21 1981-01-20    15.5
```

A line plot of the time series is also created.

2. Split Dataset

We can split the dataset into two parts.

The first part is the training dataset that we will use the test dataset that we will pretend is not available. It ple.

The dataset contains data from January 1st 1981 t

We will hold back the last 7 days of the dataset fro ose time steps as out of sample.

Specifically 1990-12-25 to 1990-12-31:

```
1 1990-12-25,12.9
2 1990-12-26,14.6
3 1990-12-27,14.0
4 1990-12-28,13.6
5 1990-12-29,13.5
6 1990-12-30,15.7
7 1990-12-31,13.0
```

The code below will load the dataset, split it into the training and validation datasets, and save them to files *dataset.csv* and *validation.csv* respectively.

```
# split the dataset
from pandas import read_csv
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0)
split_point = len(series) - 7
dataset, validation = series[0:split_point], series[split_point:]
print('Dataset %d, Validation %d' % (len(dataset), len(validation)))
dataset.to_csv('dataset.csv', index=False)
validation.to_csv('validation.csv', index=False)
```

Run the example and you should now have two files to work with.

The last observation in the dataset.csv is Christmas Eve 1990:

```
1 1990-12-24,10.0
```

That means Christmas Day 1990 and onwards are out-of-sample time steps for a model trained on *dataset.csv*.

3. Develop Model

In this section, we are going to make the data stationary and develop a simple ARIMA model.

The data has a strong seasonal component. We can neutralize this and make the data stationary by taking the seasonal difference. That is, we can take the observation for a day and subtract the observation from the same day one year ago.

This will result in a stationary dataset from which we can fit a model.

```
# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - in
        diff.append(value)
    return numpy.array(diff)
```

We can invert this operation by adding the value o
this to any forecasts made by a model trained on t

```
# invert differenced value
def inverse_difference(history, yhat, inter
    return yhat + history[-interval]
```

We can fit an ARIMA model.

Fitting a strong ARIMA model to the data is not the focus of this post, so rather than going through the analysis of the problem or grid searching parameters, I will choose a simple ARIMA(7,0,7) configuration.

We can put all of this together as follows:

```
from pandas import read_csv
from statsmodels.tsa.arima_model import ARIMA
import numpy

# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return numpy.array(diff)

# load dataset
series = read_csv('dataset.csv', header=None)
# seasonal difference
X = series.values
days_in_year = 365
differenced = difference(X, days_in_year)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit(disp=0)
# print summary of fit model
print(model_fit.summary())
```

Running the example loads the dataset, takes the seasonal difference, then fits an ARIMA(7,0,7) model and prints the summary of the fit model.

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 3278
Model:                     ARMA(7, 1)   Log Likelihood               -8673.748
Method:                       css-mle   S.D. of innovations              3.411
Date:                Mon, 20 Feb 2017   AIC                          17367.497
Time:                        10:28:38   BIC                          17428.447
Sample:                             0   HQIC                         17389.322

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0132      0.132      0.100      0.921      -0.246       0.273
ar.L1.y        1.1424      0.287      3.97
ar.L2.y       -0.4346      0.154     -2.82
ar.L3.y        0.0961      0.042      2.28
ar.L4.y        0.0125      0.029      0.43
ar.L5.y       -0.0101      0.029     -0.34
ar.L6.y        0.0119      0.027      0.44
ar.L7.y        0.0089      0.024      0.36
ma.L1.y       -0.6157      0.287     -2.14
                                    Roots
=============================================
                 Real          Imaginary
---------------------------------------------
AR.1            1.2234           -0.0000j
AR.2            1.2561           -1.0676j
AR.3            1.2561           +1.0676j
AR.4            0.0349           -2.0160j
AR.5            0.0349           +2.0160j
AR.6           -2.5770           -1.3110j
AR.7           -2.5770           +1.3110j
MA.1            1.6242           +0.0000j
-----------------------------------------------------------------------------
```


We are now ready to explore making out-of-sample forecasts with the model.

4. One-Step Out-of-Sample Forecast

ARIMA models are great for one-step forecasts.

A one-step forecast is a forecast of the very next time step in the sequence from the available data used to fit the model.

In this case, we are interested in a one-step forecast of Christmas Day 1990:

```
1990-12-25
```

Forecast Function

The statsmodel ARIMAResults object provides a *forecast()* function for making predictions.

By default, this function makes a single step out-of-sample forecast. As such, we can call it directly and make our forecast. The result of the *forecast()* function is an array containing the forecast value, the standard error of the forecast, and the confidence interval information. Now, we are only interested in the first element of this forecast, as follows.

```
# one-step out-of sample forecast
forecast = model_fit.forecast()[0]
```

Once made, we can invert the seasonal difference and convert the value back into the original scale.

```
# invert the differenced forecast to something usable
forecast = inverse_difference(X, forecast, days_in_year)
```

The complete example is listed below:

```
from pandas import read_csv
from statsmodels.tsa.arima_model import ARIMA
import numpy

# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - i
        diff.append(value)
    return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, inte
    return yhat + history[-interval]

# load dataset
series = read_csv('dataset.csv', header=No
# seasonal difference
X = series.values
days_in_year = 365
differenced = difference(X, days_in_year)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit(disp=0)
# one-step out-of sample forecast
forecast = model_fit.forecast()[0]
# invert the differenced forecast to something usable
forecast = inverse_difference(X, forecast, days_in_year)
print('Forecast: %f' % forecast)
```


Running the example prints 14.8 degrees, which is close to the expected 12.9 degrees in the *validation.csv* file.

```
Forecast: 14.861669
```

Predict Function

The statsmodel ARIMAResults object also provides a *predict()* function for making forecasts.

The predict function can be used to predict arbitrary in-sample and out-of-sample time steps, including the next out-of-sample forecast time step.

The predict function requires a start and an end to be specified, these can be the indexes of the time steps relative to the beginning of the training data used to fit the model, for example:

```
# one-step out of sample forecast
start_index = len(differenced)
end_index = len(differenced)
forecast = model_fit.predict(start=start_index, end=end_index)
```

The start and end can also be a datetime string or

```
start_index = '1990-12-25'
end_index = '1990-12-25'
forecast = model_fit.predict(start=start_index, end=end_index)
```

and

```
from pandas import datetime
start_index = datetime(1990, 12, 25)
end_index = datetime(1990, 12, 26)
forecast = model_fit.predict(start=start_index, end=end_index)
```

Using anything other than the time step indexes results in an error on my system, as follows:

```
AttributeError: 'NoneType' object has no attribute 'get_loc'
```

Perhaps you will have more luck; for now, I am stic

The complete example is listed below:

```
from pandas import read_csv
from statsmodels.tsa.arima_model import AR
import numpy
from pandas import datetime

# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - i
        diff.append(value)
    return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, inte
    return yhat + history[-interval]

# load dataset
series = read_csv('dataset.csv', header=None)
# seasonal difference
X = series.values
days_in_year = 365
differenced = difference(X, days_in_year)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit(disp=0)
# one-step out of sample forecast
start_index = len(differenced)
end_index = len(differenced)
forecast = model_fit.predict(start=start_index, end=end_index)
# invert the differenced forecast to something usable
forecast = inverse_difference(X, forecast, days_in_year)
print('Forecast: %f' % forecast)
```

Running the example prints the same forecast as above when using the *forecast()* function.

```
Forecast: 14.861669
```

You can see that the predict function is more flexible. You can specify any point or contiguous forecast interval in or out of sample.

Now that we know how to make a one-step forecast, we can now make some multi-step forecasts.

5. Multi-Step Out-of-Sample F

We can also make multi-step forecasts using the *forecast()* and *predict()* functions.

It is common with weather data to make one week (7-day) forecasts, so in this section we will look at predicting the minimum daily temperature for the next 7 out-of-sample time steps.

Forecast Function

The *forecast()* function has an argument called *steps* that allows you to specify the number of time steps to forecast.

By default, this argument is set to 1 for a one-step out-of-sample forecast. We can set it to 7 to get a forecast for the next 7 days.

```
# multi-step out-of-sample forecast
forecast = model_fit.forecast(steps=7)[0]
```

We can then invert each forecasted time step, one the forecast value for t+2, we need the inverted foreca a list called history for use when calling *inverse_diffe*

```
# invert the differenced forecast to someth
history = [x for x in X]
day = 1
for yhat in forecast:
    inverted = inverse_difference(history,
    print('Day %d: %f' % (day, inverted))
    history.append(inverted)
    day += 1
```

The complete example is listed below:

```
from pandas import read_csv
from statsmodels.tsa.arima_model import ARIMA
import numpy

# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
    return yhat + history[-interval]

# load dataset
series = read_csv('dataset.csv', header=None)
# seasonal difference
X = series.values
days_in_year = 365
differenced = difference(X, days_in_year)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit(disp=0)
# multi-step out-of-sample forecast
forecast = model_fit.forecast(steps=7)[0]
# invert the differenced forecast to something usable
history = [x for x in X]
day = 1
for yhat in forecast:
```

```
    inverted = inverse_difference(history, yhat, days_in_year)
    print('Day %d: %f' % (day, inverted))
    history.append(inverted)
    day += 1
```

Running the example prints the forecast for the next 7 days.

```
Day 1: 14.861669
Day 2: 15.628784
Day 3: 13.331349
Day 4: 11.722413
Day 5: 10.421523
Day 6: 14.415549
Day 7: 12.674711
```

Predict Function

The *predict()* function can also forecast the next 7

Using time step indexes, we can specify the end ir le:

```
# multi-step out-of-sample forecast
start_index = len(differenced)
end_index = start_index + 6
forecast = model_fit.predict(start=start_in
```

The complete example is listed below.

```
from pandas import read_csv
from statsmodels.tsa.arima_model import AR
import numpy

# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
    return yhat + history[-interval]

# load dataset
series = read_csv('dataset.csv', header=None)
# seasonal difference
X = series.values
days_in_year = 365
differenced = difference(X, days_in_year)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit(disp=0)
# multi-step out-of-sample forecast
start_index = len(differenced)
end_index = start_index + 6
forecast = model_fit.predict(start=start_index, end=end_index)
# invert the differenced forecast to something usable
history = [x for x in X]
day = 1
for yhat in forecast:
    inverted = inverse_difference(history, yhat, days_in_year)
    print('Day %d: %f' % (day, inverted))
    history.append(inverted)
    day += 1
```

Running the example produces the same results as calling the *forecast()* function in the previous section, as you would expect.

```
Day 1: 14.861669
Day 2: 15.628784
Day 3: 13.331349
Day 4: 11.722413
Day 5: 10.421523
Day 6: 14.415549
Day 7: 12.674711
```

Summary

In this tutorial, you discovered how to make out-of-sample forecasts in Python using statsmodels.

Specifically, you learned:

- How to make a one-step out-of-sample foreca
- How to make a 7-day multi-step out-of-sample
- How to use both the *forecast()* and *predict()* fu

Do you have any questions about out-of-sample fo the comments and I will do my best to answer.

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

View all posts by Jason Brownlee →

‹ Time Series Forecasting with Python 7-Day Mini-Course

Sensitivity Analysis of History Size to Forecast Skill with ARIMA in Python ›

184 Responses to *How to Make Out-of-Sample Forecasts with ARIMA in Python*

Steve March 24, 2017 at 10:44 pm #

Your tutorials are the most helpful machin
have been hugely helpful in work and personal sid
love to see a series of posts on recommender syst

Jason Brownlee March 25, 2017 at 7:36

Thanks Steve, and great suggestion!

Tim April 27, 2017 at 12:43 pm #

Hi,

This is a really nice example. Do you know if the ARIMA class allows to define the specification of the model without going through the fitting procedure. Let's say I have parameters that were estimated using a dataset that I no longer have but I still want to produce a forecast.

Thanks

Jason Brownlee April 28, 2017 at 7:32 am #

REPLY

I expect you can set the coefficients explicitly within the ARIMA model.

Sorry I do not have an example, this post may be relevant:
http://machinelearningmastery.com/make-manual-predictions-arima-models-python/

masum May 11, 2017 at 8:32 pm #

REPLY

sir,

would it be possible to do the same using LSTM R

if it is would you please come up with a blog?

Thanking you

REPLY

Jason Brownlee May 12, 2017 at 7:41 am #

Yes.

Any of my LSTM tutorials show how to make out of sample forecasts. For example: http://machinelearningmastery.com/multi-step-time-series-forecasting-long-short-term-memory-networks-python/

masum May 12, 2017 at 8:29 pm #

I tried to run the above example without a

```
from pandas import Series
from matplotlib import pyplot
from pandas import Series
from statsmodels.tsa.arima_model import ARIMA
# load dataset
series = Series.from_csv('daily-minimum-temperat
print(series.head(20))
series.plot()
pyplot.show()

split_point = len(series) – 7
dataset, validation = series[0:split_point], series[split_point:]
print('Dataset %d, Validation %d' % (len(dataset), len(validation)))
dataset.to_csv('dataset.csv')
validation.to_csv('validation.csv')

series = Series.from_csv('dataset.csv', header=None)
model = ARIMA(series, order=(7,0,1))
model_fit = model.fit(disp=0)

forecast = model_fit.forecast(steps=7)[0]
print('Forecast: %f' % forecast)
```

for the code i am getting an error:

TypeError: only length-1 arrays can be converted to Python scalars

how can i solve this? it does well for single step forecast

REPLY

Jason Brownlee May 13, 2017 at 6:13 am #

I would recommend double checking your data, make sure any footer information was deleted.

Hans June 1, 2017 at 12:58 am #

REPLY

What does ‘seasonal difference’ mean?

And what are the details of:

‘Once made, we can invert the seasonal difference and convert the value back into the original scale.’

Is it worth to test this code with non-seasonal data or is there another ARIMA-tutorial for non-seasonal approaches on this site?

Jason Brownlee June 2, 2017 at 12:51 p

See this post:
http://machinelearningmastery.com/seasonal-p

And this post:
http://machinelearningmastery.com/time-series

Please use the search feature of the blog.

Hans June 15, 2017 at 11:27 am #

If I pretend data in test-partition is not give
seasonal cleaning?

http://machinelearningmastery.com/tune-arima-parameters-python/

Manjunath b March 4, 2020 at 6:01 am #

REPLY

Hi Jason really it was great article
I have one doubt say when future data coming from weather station due to some fault values are missing if we randomly miss some data from sensor then I need to fill it using ARIMA by using prediction method
But here start and end date parameter is required so can I pass only start date and end date can I left it blank is it works ?

Jason Brownlee March 4, 2020 at 6:02 am #

REPLY

Perhaps experiment and see what works best for your use case.

Hans June 15, 2017 at 11:29 am #

REPLY

Can I obtain a train RMSE from this example. Is training involved?

Jason Brownlee June 16, 2017 at 7:47 am #

REPLY

The model is trained, then the trained model is used to make a forecast.

Consider reading and working through the tutorial.

Hans June 16, 2017 at 12:16 pm #

I did so several times.
How can I obtain a train RMSE from the m

Jason Brownlee June 17, 201

See this post on how to estim
sample predictions:
http://machinelearningmastery.com/ba
forecasting/

See this post to understand the difference between evaluating a model and using a final model to make predictions:
http://machinelearningmastery.com/train-final-machine-learning-model/

Hans June 19, 2017 at 5:35 am #

I actually meant obtain a train RMSE from the model in the example.
As I understand the model was trained before making an out of sample prediction.
If we place a

print(model_fit.summary())

right after fitting/training it prints some information's, but no train RMSE.

A)
Is there a way to use the summery-information to obtain a train RMSE?
B)
Is there a way in Python to obtain all properties and methods from the model_fit object-like in other languages?

Jason Brownlee June 19, 2017 at 8:47 am #

Yes, this tutorial assumes you have already estimated the skill of your model and are now ready to use it to make forecasts.

Estimating the skill of the model is a different task. You can do this using walk forward validation or a train/test split evaluation.

REPLY

Hans June 16, 2017 at 3:06 pm #

Is this the line where the training h

model = ARIMA(differenced, order=(7,0,1))

Jason Brownlee June 17, 201

No here:

```
model_fit = model.fit(disp=0)
```

Hans June 25, 2017 at 12:29 pm #

Yes I know. I actually thought there could be a direct answer to A) and B). I would use it for archiving.

REPLY

Hans June 15, 2017 at 12:40 pm #

If I write: ‘split_point = len(series) – 0’ while my last datapoint in dataset is from today.

Would I have a valid forecast for tomorrow?

REPLY

M.Swefy June 22, 2017 at 12:39 am #

thanks a lot for the nice detailed article, i followed all steps and they all seem working properly, i seek your support Dr. to help me organize my project.

i have a raw data for temperature readings for some nodes (hourly readings), i selected the training set and divided them to test and training sets.
i used ARINA model to train and test and i got Test MSE: 3.716.

now i need to expose the mass raw data to the trained model, then get the forecased values vs. the actual values in the same csv file.

what should i do

M.Swefy June 22, 2017 at 12:41 am #

REPLY

*ARIMA

Jason Brownlee June 22, 2017 at 6:09 am #

REPLY

I'm not sure I follow. Consider this post on how to evaluate a time series model:
http://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

AMU June 23, 2017 at 5:33 am #

Thank you Jason for this wonderful post…

Do you also have something similar for LSTM Neu
to Make Out-of-Sample Forecasts with LSTM in Py

If not, will you write one blog like this with detail ex
same question.

Jason Brownlee June 23, 2017 at 6:45 a

Almost every post I have on LSTMs shows how to make out of sample forecasts. The code is wrapped up in the walk-forward validation.

Franklin July 1, 2017 at 1:09 am #

REPLY

Hi Jason,

Thanks a lot for this lesson. It was pretty straightforward and easy to follow. It would have been a nice bonus to show how to evaluate the forecasts though with standard metrics. We separated the validation set out and forecasted values for that week, but didn't compare to see how accurate the forecast was.

On that note, I want to ask, does it make sense to use R^2 to score a time series forecast against test data? I'm trying to create absolute benchmarks for a time series that I'm analyzing and want to report unit-independent metrics, i.e. not standard RMSE that is necessarily expressed in the problem's unit scale. What about standardizing the data using zero mean and unit variance, fitting ARIMA, forecasting, and reporting that RMSE? I've been doing this and taking the R^2 and the results are pretty interpretable. RMSE: 0.149 / R^2: 0.8732, but I'm just wondering if doing things this way doesn't invalidate something along the way. Just want to be correct in my process.

Thanks!

Jason Brownlee July 1, 2017 at 6:37 am #

REPLY

We do that in other posts. Tens of other posts in fact.

This post was laser focused on “how do I make a prediction when I don’t know the real answer”.

Yes, if R^2 is meaningful to you, that you can interpret it in your domain.

Generally, I recommend inverting all transforms on the prediction and then evaluating model skill at least for RMSE or MAE where you want apples-to-apples. This may be less of a concern for an R^2.

Vishanth July 19, 2017 at 6:56 am #

Seriously amazing. Thanks a lot professor

Jason Brownlee July 19, 2017 at 8:30 ar

Thanks. Also, I’m not a professor.

Kirui July 20, 2017 at 5:15 pm #

I get this error from your code

Traceback (most recent call last):
File “..”, line 22, in
differenced = difference(X, days_in_year)
File “..”, line 9, in difference
value = dataset[i] – dataset[i – interval]
TypeError: unsupported operand type(s) for -: ‘str’ and ‘str’

Cant tell where the problem is.

Jason Brownlee July 21, 2017 at 9:31 am #

REPLY

Perhaps check that you have loaded your data correct (as real values) and that you have copied all of the code from the post without extra white space.

Antoine August 23, 2017 at 1:00 am #

REPLY

Hi Jason,
Thanks for this detailled explanation. Very clear.

Do you know if it is possible to use the fitted parameters of an ARMA model (ARMAResults.params) and apply it on an other data set ?

I have an online process that compute a forecasting and I would like to have only one learning process (one usage of the fit() function). The rest of the time, I would like to applied the previously found parameters to the data.

Thanks in advance !

REPLY

Jason Brownlee August 23, 2017 at 6:56 am #

Yes, you can use a grid search:
https://machinelearningmastery.com/grid-searc

Bob October 6, 2017 at 11:53 pm #

Ciao Jason,
Thanks for this tutorial and all the time series relate
write both posts and code.
I'm by the way still confused about something whic
The ARIMA parameters specify the lag which it use
In your case you used p=7 for example so that you
A first silly question is why do I need to fit an entire
The second question is that fitting my model I get a
training (2 days vs 1 year) which would reinforce my first point.
What am I missing?
Thanks

REPLY

Jason Brownlee October 7, 2017 at 5:56 am #

The model needs lots of examples in order to generalize to new cases.

More data is often better, to a point of diminishing returns in terms of model skill.

REPLY

Kai October 31, 2017 at 12:02 pm #

Hi Jason. Thanks for this awesome post.
But I have a question that is it possible to fit a multivariable time series using ARIMA model? Let's say we have a 312-dimension at each time step in the dataset.
Thanks!

Jason Brownlee October 31, 2017 at 2:5

Yes, but you will need to use an extension of ARIMA called ARIMAX. I do not have an example, sorry.

Dave J November 5, 2017 at 7:12 am #

REPLY

Hi Dr Brownlee, thanks so much for the tutorials!

I've searched but didn't find anyhting – perhaps my fault…

But do you have any tutorials or suggestions about forecasting with limited historical observations? Specifically, I'm in a position where some sensors may have a very limited set of historical observations (complete, but short, say it's only been online for a
possibly be used as historical analogies (multiple y

I've considered constructing a process that uses ea
iterating over each sensor and finding which senso
sensors.

However I'm struggling to find any established bes
suggestions for me?

If not I understand, but I really appreciate all the ins
book!

Jason Brownlee November 6, 2017 at 4:

Great question.

You might be able to use the historical data or models for different but similar sensors (one some dimension). Get creative!

Dave J November 6, 2017 at 10:53 am #

REPLY

I would likely just be looking at the RMSE and MAE to gauge accuracy, correct? Is there another measure of fitness I would be wise to consider?

Jason Brownlee November 7, 2017 at 9:45 am #

REPLY

No MSE and RMSE are error scores for regression problems. Accuracy is for classification problems (predicting a label).

Debola November 11, 2017 at 5:28 am #

REPLY

Hi, Geat tutorial. A question about the diff

Jason Brownlee November 11, 2017 at 9:24 am #

REPLY

It doesn't, that would be a good extension to this tutorial.

Debola November 12, 2017 at 12:37 am #

REPLY

Is it possible to apply seasonal_decompose on the dataset used in this tutorial since it's a daily forecast. Most applications of seasonal_decompose i have seen are usually on monthly and quarterly data

Jason Brownlee November 12

Yes, you could use it on this d

Akanksha November 19, 2017 at 4:32 am #

Thank you for an amazing tutorial. I wante
predicted in the end of your tutorial into a variable f

Jason Brownlee November 19, 2017 at 11:10 am #

REPLY

Sure, you can assign them to a variable or save them to file.

Jonathon July 29, 2018 at 10:45 am #

REPLY

Thank you for the amazing blog!, I am finding it difficult to assign multi-step values to variable, Could you please help me with the same.

Thanks in Advance!

Jason Brownlee July 30, 2018 at 5:43 am #

REPLY

What is the problem exactly?

Kapil July 29, 2018 at 10:36 pm #

Hi Jason, Thank you for the amazing blog, could you please help me with assigning multi-step predict values to variable.

REPLY

Jason Brownlee July 30, 2018 at 5:48 am #

You can use the forecast() function and specify the number of steps.

REPLY

kapil August 8, 2018 at 2:31 am #

Thank you for your response
function and with predict() function, Pre
them to assigned to variable, Can that

Thanks in Advance!

Jason Brownlee August

That is surprising, if not in

Perhaps confirm that you are provi

Kapil August 8, 2018 at 6:56 am #

No Worries, I got it – Thank you

REPLY

Satyajit Pattnaik December 21, 2017 at 5:01 pm #

@Jason, Thanks for this, but my dataset is in a different format, it's in YYYY-MM-DD HH:MI:SS, and the data is hourly data, let say if we have data till 11/25/2017 23:00 5.486691952

And we need to predict the next day's data, so we need to predict our next 24 steps, what needs to be done?

Need a help on this.

REPLY

Jason Brownlee December 22, 2017 at 5:31 am #

Sure, you can specify the date-time format when loading the Pandas Series.

You can predict multiple steps using the predict() function.

Satyajit Pattnaik December 21, 2017 at 8:02 pm #

REPLY

One more question on top of my previous question,
let say my data is hourly data, and i have one week's data as of now, as per your code do i have to take the days_in_year parameter as 7 for my case?

And as per my data's ACF & PACF, my model should be ARIMA(xyz, order=(4,1,2))
and taking the days_in_year parameter as 7, is giving my results, but not sure how correct is that..
please elaborate a bit @Jason

Jason Brownlee December 22, 2017 at 5

I would recommend tuning the model

Satyajit Pattnaik January 3, 2018 at 11:47 pm

Hi Jason,

I am bugging you, but here's my last question, my
as per the ACF, PACF plots.

Now my code looks like this:

```
history = [x for x in train]
predictions = list()
for t in range(len(test)):
    model = ARIMA(history, order=(6,1,2))
    model_fit = model.fit(disp=0)
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
```

Here, as i am appending obs to the history data, what if i add my prediction to history and then pass it to the model, do i have to run this in a loop to predict pdq values again in a loop?

My question is, if we are doing Recursive multi step forecast do we have to run the history data to multiple ARIMA models, or can we just use history.append(yhat) in the above code and get my results?

Jason Brownlee January 4, 2018 at 8:12 am #

REPLY

Recursive multi-step means you will use predictions as history when you re-fit the model.

Satyajit Pattnaik January 4, 2018 at 4:48 pm #

REPLY

Reply to my previous response, s
be doing history.append(yhat) instead of hi

code using the same ARIMA model i.e. 6,1,2 or for each history we will determine the pdq values and run on multiple ARIMA models to get the next predictions?

I hope, you are getting my point.

REPLY

Jason Brownlee January 5, 2018 at 5:19 am #

It is up to you.

Olagot Andree January 7, 2018 at 1:06 pm #

Hello,
I am actually working on a project for implicit volatil
tutorial has been a lot of help but i just want to clari
1. Is it okay to train on the all dataset and not divid
2. What is the sample of data selected for the forec
original dataset?

Thank you

Jason Brownlee January 8, 2018 at 5:40

You must evaluate the skill of your mo
that with time series:
https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

REPLY

Sooraj February 2, 2018 at 2:04 pm #

How do we add more input parameters? Like for example, i would like to predict the weather forecast based on historic forecast but i would also like to consider, say the total number of rainy days last 10 years and have both influence my prediction?

REPLY

Jason Brownlee February 3, 2018 at 8:32 am #

You may have to use a different linear model such as ARIMAX.

REPLY

Sooraj February 7, 2018 at 9:13 am #

Thank you.

Do you have any samples that I could learn from or use as a base to build my own forecast? Similar to the article that you shared above?

Jason Brownlee February 7, 2018 at 9:34 am #

REPLY

Perhaps try searching the blog and see if there is a tutorial that is a good fit?

Sooraj February 19, 2018 at 6:55 am #

Will do that. Thanks!

Daphne February 5, 2018 at 1:51 am #

Hey Jason, let's say if I wanted to forecast

change the line below to:

forecast = model_fit.forecast(steps=365)[0]

Will it works? Thanks!

Jason Brownlee February 5, 2018 at 7:4

Yes, but expect skill to be very poor.

Daphne February 5, 2018 at 2:11 pm #

REPLY

Thanks so much Jason! But just a quick check with you, why are you splitting the dataset into two different csv: dataset.csv and validation.csv? What is the purpose each of the csv?

Jason Brownlee February 5, 2018 at 2:53 pm #

REPLY

This post might clear things up:
https://machinelearningmastery.com/difference-test-validation-datasets/

Chuck February 18, 2018 at 12:23 pm #

REPLY

Hi Jason,

Thank you for sharing a such wonderful article with us which I am looking for a while.

However, I got an error of “ValueError: The computed initial AR coefficients are not stationary.” when run your code block 5 beneath “We can put all of this together as follows:”

If I run it under Sypder, I got “cannot import name ‘recarray_select’”.

It would be appreciated if you could give me some clue how to fix it.

Thank you!

Chuck

Jason Brownlee February 19, 2018 at 9:

Was this with the data provided in the

You can learn more about stationarity here:
https://machinelearningmastery.com/time-serie

masum March 9, 2018 at 12:59 pm #

how can we calculate the total RMSE?

Jason Brownlee March 10, 2018 at 6:16

The square root of the mean squared differences between predicted and expected values.

Rishabh Agrawal March 30, 2018 at 3:19 am #

REPLY

Hi Jason,

Thanks for the wonderful post.

One thing which I can’t understand is that we are forecasting for the next 7 days in the same dataset (dataset.csv) that we have trained the model on.

In other words, in the initial steps we had split the data into ‘dataset.csv’ and ‘validation.csv’ and then we fit the ARIMA on ‘dataset.csv’ but we never called ‘validation.csv’ before making a forecast. How does it wok?

Jason Brownlee March 30, 2018 at 6:44 am #

REPLY

No, we are forecasting beyond the end of dataset.csv as though validation.csv does not exist. We can then look in validation.csv and se

Perhaps re-read the tutorial?

REPLY

Rishabh Agrawal March 30, 2018 at 5:04 pm #

yep! got it. Actually I have exogenous inputs as well. So, I had to use 'validation' dataset as well.

REPLY

Jason Brownlee March 31, 2018 at 6:34 am #

Great.

aadi April 19, 2018 at 9:14 pm #

Hi jason
Can you tell why did we leave the test data as it is?
and what if so in the above method we dont separa

Jason Brownlee April 20, 2018 at 5:49 a

In the above tutorial we are pretendin
we do not know the true outcome values.

REPLY

Serkan May 17, 2018 at 6:34 pm #

Could you please tell about what should be changed in the code if multivariate analysis is done, i.e, if we have extra 3 features in dataset.

REPLY

Jason Brownlee May 18, 2018 at 6:21 am #

Different methods will need to be used. I hope to have examples soon.

REPLY

Piyasi Choudhury May 30, 2018 at 8:27 am #

Hi Jason, Thanks for the post..very intuitive. I am at Step3: Developing Model. I ran through the other doc on: how to choose your grid params for ARIMA configuration and came up with (10,0,0) with the lowest MSerror. I do the following:

```
# seasonal difference
X = series.values
days_in_year = 365
differenced = difference(X, days_in_year)

# fit model
model = ARIMA(differenced, order=(10,0,0))
```

and get error: Insufficient degrees of freedom to estimate.

My data is on monthly level (e.g. 1/31/2014, 2/28/2014, 3/31/2014)..I have 12 readings from each year of 2014-2017+3 readings from 2018 making it 52 readings. Do I have to change the #seasonal difference based on this?

Thanks

Jason Brownlee May 30, 2018 at 3:07 pm

It is a good idea to seasonally adjust i
via SARIMA.

vamshi December 4, 2018 at 9:24 pm #

i am getting same problem what shou

SJ June 17, 2018 at 6:00 am # REPLY

@ Jason

Thank you for your article, this is helpful.
I used Shampo sales dataset and used ARIMA Forecast & Predict function for next 12 months but i get different results.

Jason Brownlee June 18, 2018 at 6:36 am # REPLY

Perhaps you have done something different to the tutorial?

Rasangika June 23, 2018 at 8:42 pm # REPLY

Hello sir,

Can you please tell me how i can take the predicted output to a CSV ?
Thank you!

Jason Brownlee June 24, 2018 at 7:32 am #

REPLY

You can save an array as a CSV File via numpy.

https://docs.scipy.org/doc/numpy-1.14.0/reference/generated/numpy.savetxt.html

Kay July 10, 2018 at 6:34 am #

REPLY

Hi, @Jason
I am trying to use predict(start, end), and I found only integer parameter will work. I want to specify the start and end by a date, but it gives me an error:
'only integers, slices (`:`), ellipsis (`...`), numpy.new
indices'
I have searched a lot online, but none of them wor

Jason Brownlee July 10, 2018 at 6:54 a

The API says it does support dates, a
have not tried it though, sorry.

Shivaprasad July 20, 2018 at 5:23 pm #

If my dataset is less than 365 days it is showng an error in the below code:If my dataset is of just 50rows how that can be perfomed?

```
from pandas import Series
from statsmodels.tsa.arima_model import ARIMA
import numpy

# create a differenced series
def difference(dataset, interval=1):
diff = list()
for i in range(interval, len(dataset)):
value = dataset[i] – dataset[i – interval]
diff.append(value)
return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
return yhat + history[-interval]

# load dataset
series = Series.from_csv('dataset.csv', header=None)
# seasonal difference
X = series.values
days_in_year = 365
```

```
differenced = difference(X, days_in_year)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit(disp=0)
# multi-step out-of-sample forecast
forecast = model_fit.forecast(steps=7)[0]
# invert the differenced forecast to something usable
history = [x for x in X]
day = 1
for yhat in forecast:
inverted = inverse_difference(history, yhat, days_in_year)
print('Day %d: %f' % (day, inverted))
history.append(inverted)
day += 1
```

Jason Brownlee July 21, 2018 at 6:30 ar

Sorry, I cannot debug your example.

Fel September 2, 2018 at 8:23 am #

I am trying to apply this code to other data

```
C:\Users\Fel\Anaconda3\lib\site-packages\statsmo
zero encountered in true_divide
invmacoefs = -np.log((1-macoefs)/(1+macoefs))
C:\Users\Fel\Anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:650: RuntimeWarning: invalid value encountered in true_divide
newparams = ((1-np.exp(-params))/(1+np.exp(-params))).copy()
C:\Users\Fel\Anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:651: RuntimeWarning: invalid value encountered in true_divide
tmp = ((1-np.exp(-params))/(1+np.exp(-params))).copy()
---------------------------------------------------------------------------
LinAlgError Traceback (most recent call last)
in ()
24 # fit model
25 model = ARIMA(differenced, order=(7,0,1))
—> 26 model_fit = model.fit(disp=0)
27 # multi-step out-of-sample forecast
28 forecast = model_fit.forecast(steps=period_forecast)[0]

~\Anaconda3\lib\site-packages\statsmodels\tsa\arima_model.py in fit(self, start_params, trend, method, transparams, solver, maxiter, full_output, disp, callback, start_ar_lags, **kwargs)
957 maxiter=maxiter,
958 full_output=full_output, disp=disp,
–> 959 callback=callback, **kwargs)
```

960 params = mlefit.params
961

~\Anaconda3\lib\site-packages\statsmodels\base\model.py in fit(self, start_params, method, maxiter, full_output, disp, fargs, callback, retall, skip_hessian, **kwargs)
464 callback=callback,
465 retall=retall,
–> 466 full_output=full_output)
467
468 # NOTE: this is for fit_regularized and should be generalized

~\Anaconda3\lib\site-packages\statsmodels\base\optimizer.py in _fit(self, objective, gradient, start_params, fargs, kwargs, hessian, method, maxiter, full_output, disp, callback, retall)
189 disp=disp, maxiter=maxiter, callback=callback,
190 retall=retall, full_output=full_output,
–> 191 hess=hessian)
192
193 optim_settings = {‘optimizer’: method, ‘start_pa

~\Anaconda3\lib\site-packages\statsmodels\base\c
kwargs, disp, maxiter, callback, retall, full_output, h
408 callback=callback, args=fargs,
409 bounds=bounds, disp=disp,
–> 410 **extra_kwargs)
411
412 if full_output:

~\Anaconda3\lib\site-packages\scipy\optimize\lbfgs
approx_grad, bounds, m, factr, pgtol, epsilon, iprint
197
198 res = _minimize_lbfgsb(fun, x0, args=args, jac=jac, bounds=bounds,
–> 199 **opts)
200 d = {‘grad’: res[‘jac’],
201 ‘task’: res[‘message’],

~\Anaconda3\lib\site-packages\scipy\optimize\lbfgsb.py in _minimize_lbfgsb(fun, x0, args, jac, bounds, disp, maxcor, ftol, gtol, eps, maxfun, maxiter, iprint, callback, maxls, **unknown_options)
333 # until the completion of the current minimization iteration.
334 # Overwrite f and g:
–> 335 f, g = func_and_grad(x)
336 elif task_str.startswith(b’NEW_X’):
337 # new iteration

~\Anaconda3\lib\site-packages\scipy\optimize\lbfgsb.py in func_and_grad(x)
278 if jac is None:
279 def func_and_grad(x):
–> 280 f = fun(x, *args)
281 g = _approx_fprime_helper(x, fun, epsilon, args=args, f0=f)
282 return f, g

~\Anaconda3\lib\site-packages\scipy\optimize\optimize.py in function_wrapper(*wrapper_args)
291 def function_wrapper(*wrapper_args):
292 ncalls[0] += 1

Start Machine Learning

```
–> 293 return function(*(wrapper_args + args))
294
295 return ncalls, function_wrapper

~\Anaconda3\lib\site-packages\statsmodels\base\model.py in f(params, *args)
438
439 def f(params, *args):
–> 440 return -self.loglike(params, *args) / nobs
441
442 if method == 'newton':

~\Anaconda3\lib\site-packages\statsmodels\tsa\arima_model.py in loglike(self, params, set_sigma2)
778 method = self.method
779 if method in ['mle', 'css-mle']:
–> 780 return self.loglike_kalman(params, set_sigr
781 elif method == 'css':
782 return self.loglike_css(params, set_sigma2)

~\Anaconda3\lib\site-packages\statsmodels\tsa\ari
set_sigma2)
788 Compute exact loglikelihood for ARMA(p,q) m
789 """
–> 790 return KalmanFilter.loglike(params, self, se
791
792 def loglike_css(self, params, set_sigma2=True

~\Anaconda3\lib\site-packages\statsmodels\tsa\kal
arma_model, set_sigma2)
647 loglike, sigma2 = kalman_loglike.kalman_loglik
648 k_ar, k_ma, k_lags, int(nobs), Z_mat,
–> 649 R_mat, T_mat)
650 elif issubdtype(paramsdtype, np.complex128):
651 loglike, sigma2 = kalman_loglike.kalman_loglike_complex(y, k,

kalman_loglike.pyx in statsmodels.tsa.kalmanf.kalman_loglike.kalman_loglike_double()

kalman_loglike.pyx in statsmodels.tsa.kalmanf.kalman_loglike.kalman_filter_double()

~\Anaconda3\lib\site-packages\numpy\linalg\linalg.py in pinv(a, rcond)
1722 return wrap(res)
1723 a = a.conjugate()
-> 1724 u, s, vt = svd(a, full_matrices=False)
1725
1726 # discard small singular values

~\Anaconda3\lib\site-packages\numpy\linalg\linalg.py in svd(a, full_matrices, compute_uv)
1442
1443 signature = 'D->DdD' if isComplexType(t) else 'd->ddd'
-> 1444 u, s, vh = gufunc(a, signature=signature, extobj=extobj)
1445 u = u.astype(result_t, copy=False)
1446 s = s.astype(_realType(result_t), copy=False)

~\Anaconda3\lib\site-packages\numpy\linalg\linalg.py in _raise_linalgerror_svd_nonconvergence(err
flag)
```


Start Machine Learning

```
96
97 def _raise_linalgerror_svd_nonconvergence(err, flag):
—> 98 raise LinAlgError("SVD did not converge")
99
100 def get_linalg_error_extobj(callback):
```

LinAlgError: SVD did not converge

Jason Brownlee September 3, 2018 at 6:09 am #

REPLY

Perhaps try some other configurations of the model?
Perhaps try to scale or difference your data firs
Perhaps try more or less data?

Tejas Haritsa V K September 7, 2018 at 8:10

Truly an outstanding work. I had been sea
functions and this made my day. Thank you for this

Do share your YouTube channel link if you have a

Jason Brownlee September 8, 2018 at 6

Thanks.

I don't make videos. Developers learn by doing, not watching.

Ashutosh Sharma September 17, 2018 at 7:09 am #

REPLY

I get this error from your code

```
Traceback (most recent call last):
File "..", line 22, in
differenced = difference(X, days_in_year)
File "..", line 9, in difference
value = dataset[i] – dataset[i – interval]
TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

Cant tell where the problem is.

Jason Brownlee September 17, 2018 at 2:07 pm #

REPLY

Ensure that you copy the complete ex

Bhadri October 1, 2018 at 3:42 am #

REPLY

Thanks Jason. this is very helpful.

When I run the original dataset, train it and test it, I get a MSE of .09 which is very good where I use (p,d,q) as 2,1,0.

My dataset contains 60 observations out of I push 12 to validation set.

When I forecasted using step=12 and did a MSE with validation set, I get a MSE of .42.
Is this expected and is it a good measure?

regards
Bhadri.

Jason Brownlee October 1, 2018 at 6:29

The idea of “good” is really problem sp
https://machinelearningmastery.com/faq/single

SN October 9, 2018 at 10:47 pm #

Hi Jason,

Thanks ever so much for this post! Your posts are all very clear and easy to follow. I cannot steady the heavily mathematical stuff, it just confuses me.

I have a question. If my daily data is for Mondays-Fridays, should I adjust the number of days in a year to 194 instead of 365? That is the total number of days in this year excluding holidays and weekends in Germany.

Regards,

S:N

Jason Brownlee October 10, 2018 at 6:12 am #

REPLY

It really depends if you need to seasonally adjust the data or not.

Learn more here:
https://machinelearningmastery.com/remove-trends-seasonality-difference-transform-python/

PyTom October 11, 2018 at 11:39 pm #

REPLY

Dear Jason, thank you very much for the tutorial. Is it normal that if I do a long-term prediction (for instance, 200 steps) the performance of the predictor degradates? In particular, I observe that the prediction converges to a certain value. What can I do to perform a long term out-of-sample prediction?

Jason Brownlee October 12, 2018 at 6:40 am #

REPLY

Yes, the further into the future you predict, the worse the performance. Predicting the future is very hard.

Raghu October 15, 2018 at 12:46 am #

Hi Jason, Thank you very much for the po
I checked stationarity test for the provided data-set
the result

Test Statistic -4.445747
p-value 0.000246
#Lags Used 20.000000
Number of Observation Used 3629.000000
Critical Value (1%) -3.432153
Critical Value (5%) -2.862337
Critical Value (10%) -2.567194
The result shows that data looks stationary. So my

1. Even though data is stationary why did you apply seasonality dereference ?
2. You have taken seasonality dereference of data and the parameter d of ARIMA model is still 0(ARIMA model 7 0 1). isn't required to mention d > 0(No of dereference taken) when dereference has applied on actual data?

Jason Brownlee October 15, 2018 at 7:30 am #

REPLY

The problem is easier to model with the seasonality removed.

The d parameter is intended to counter any trend, there is no trend, therefore d can remain zero.

July October 24, 2018 at 1:06 pm #

REPLY

Hi, this is wonderful.
I have a small question about the out of sample one step forecast for several days. For example, I need to predict data from 1990-12-25 to 1990-12-31, and I want to use one step forecast for every. How can I make it using api predict or forecast? Thanks.

Jason Brownlee October 24, 2018 at 2:48 pm #

REPLY

I believe the example in the tutorial above does this. Perhaps I don't understand your question?

July October 25, 2018 at 1:38 am #

REPLY

Well, thanks for the reply.
Let's talk about the 7 data from 1990-12-25 to 1990-12-31 that needs to be forecasted. In your tutorial, you use the function forecast(period=7) getting the forecasting in one time. But I want to only use the function forecast(period=1) in
forecast(period=7), the new predicted data
example, the predicted data 1990-12-25 w
forecast(period=1), every predicted data is
predicting 1990-12-26, the real data 1990-
data 1990-12-25 like in forecast(period=7).
update using statsmodels.
Forgive my unskilled expression.

Jason Brownlee October 25, 2

Ahh, I see, thanks.

I assume that real observations are made available after each prediction, so that they can be used as input.

The simplest answer is to re-fit the model with the new obs and make a 1-step prediction.

The complex answer is to study the API/code and figure out how to provide the dynamic input, I'm not sure off the cuff if the statsmodel API supports this usage.

Also, this may help for the latter:
https://machinelearningmastery.com/make-manual-predictions-arima-models-python/

July October 27, 2018 at 7:38 pm #

REPLY

Thanks for your reply again.
I have been working with the first method you mentioned. It is the correct method that can meet my demand. But it has a very high time spending. Well I test on the stock index data such as DJI.NYSE including 3000+ data. It is very hard for arima method to make a good regression. Maybe stocks data can not be predicted.

Jason Brownlee October 28, 2018 at 6:0

The stock market is not predictable:
https://machinelearningmastery.com/faq/single-faq/can-you-help-me-with-machine-learning-for-finance-or-the-stock-market

Ronak December 20, 2018 at 11:00 pm #

REPLY

Hey , I am getting error here doing import series but getting error from csv file side
Note that some of the default arguments are different, so please refer to the documentation for from_csv when changing your function calls
infer_datetime_format=infer_datetime_format)
Traceback (most recent call last):
File “sarima.py”, line 10, in
series = Series.from_csv(‘/home/techkopra/Docum
temperatures1.csv’, header=None)
File “/home/techkopra/Documents/Sarima_machin
packages/pandas/core/series.py”, line 3728, in fro
result = df.iloc[:, 0]
File “/home/techkopra/Documents/Sarima_machin
packages/pandas/core/indexing.py”, line 1472, in _
return self._getitem_tuple(key)
File “/home/techkopra/Documents/Sarima_machin
packages/pandas/core/indexing.py”, line 2013, in _
self._has_valid_tuple(tup)
File “/home/techkopra/Documents/Sarima_machin
packages/pandas/core/indexing.py”, line 222, in _h
self._validate_key(k, i)
File “/home/techkopra/Documents/Sarima_machine-learnig/env/lib/python3.6/site-packages/pandas/core/indexing.py”, line 1957, in _validate_key
self._validate_integer(key, axis)
File “/home/techkopra/Documents/Sarima_machine-learnig/env/lib/python3.6/site-packages/pandas/core/indexing.py”, line 2009, in _validate_integer

raise IndexError(“single positional indexer is out-of-bounds”)
IndexError: single positional indexer is out-of-bounds

could you support me this error ?

Thanks

Jason Brownlee December 21, 2018 at 5:29 am #

REPLY

I have some suggestions here:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Ronak December 21, 2018 at 11:40 pm #

Hey buddy,

I am getting issue this
Traceback (most recent call last):
File “hello.py”, line 23, in
differenced = difference(X, days_in_year)
File “hello.py”, line 10, in difference
value = dataset[i] – dataset[i – interval]
TypeError: unsupported operand type(s) for -: ‘str’ and ‘str’

Thanks

Jason Brownlee December 22, 2018 at 6

Are you using Python 3?

Ronak December 24, 2018 at 5:12 pm #

yes

Andy Hui February 15, 2020 at 10:15 p

it’s this project run under python 2

Jason Brownlee February 16, 2020 at 6:06 am #

REPLY

I use Python 3.6.

I expect it will work for Python 2.7.

Mayssa December 25, 2018 at 4:21 am #

REPLY

Why is it required to make the data stationary ? when you the observation for each day from the same day one year before, doesn’t this affect the data and hence the results ?

Jason Brownlee December 25, 2018 at 7:25 am #

REPLY

It greatly simplifies the prediction problem and meets the expectations of the linear model.

Try with/without and compare results!

kono February 17, 2019 at 5:59 am #

REPLY

I used your code to forecast next 365 days. But forecast values before inverse converge to 0.0131662 from 96th step on. That means forecast values after inverse are just last year's values + 0.0131662. This is almost equivalent to no forecasting at all. In real practice, how do people do forecasting for a longer future time period?

Jason Brownlee February 17, 2019 at 6:35 am #

REPLY

That is a lot of days to forecast!

From what I have seen, forecasting more than
error to be useful on most problems – it depen

kono February 17, 2019 at 9:52 am #

So normally how do people use an ARIM
it to predict next couple data points in the future? W
to update the future prediction? For example, supp
predict 2/2 to 2/10. Once 2/2 data comes in, I inclu
prediction for 2/3 to 2/10 plus 2/11. Is this the corre

Jason Brownlee February 18, 2019 at 6:27 am #

REPLY

It can be, it really depends on your production environment.

For example, in some cases, perhaps the coefficients are used directly to make a prediction, e.g. using another language. In other environments, perhaps the model can be used directly.

Also, when it comes to updating the model, I recommend testing different schedules to see what is effective for your specific data.

Mike March 6, 2019 at 9:42 am #

REPLY

Hi. How do you do this for multiple time series at the same time? for example df with 50 columns or so

Jason Brownlee March 6, 2019 at 2:46 pm #

REPLY

Perhaps one model per time series.

Start Machine Learning

Or use a method like a neural net that can support multiple input variables:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

Naveensankar March 10, 2019 at 4:42 pm #

REPLY

Hi jason, This tutorial is really awesome…
can you please help me on plotting the graph to compare the predicted and actual value and to find the RMSE score?

Jason Brownlee March 11, 2019 at 6:48

You can get started with plots here:
https://machinelearningmastery.com/load-explo

Archana March 13, 2019 at 3:58 pm #

Sir,
Your blogs were really helpful. I felt depth understa
Thank you soo much.
And I have a doubt. Can we detect Anomaly using

Jason Brownlee March 14, 2019 at 9:17 am #

REPLY

Thanks.

No, ARIMA is not really suited to anomaly detection.

kono July 14, 2019 at 6:26 am #

REPLY

“No, ARIMA is not really suited to anomaly detection.” Can you suggest some methods which are suitable for anomaly detection in time series?

Jason Brownlee July 14, 2019 at 8:17 am #

REPLY

I hope to cover this topic in great detail in the future.

Perhaps investigate the problem as an imbalanced classification task?

bipulsingh kashyap April 1, 2019 at 8:47 pm #

REPLY

I have monthly data but some months information is missing ,can i use arima on this type of data.

Jason Brownlee April 2, 2019 at 8:09 am #

REPLY

You can fill in the missing values with a mean/median value.

Bats September 26, 2019 at 5:10 am #

But what if my data has strong se

Jason Brownlee September 2

Then the value at the same p

ask April 30, 2019 at 7:04 am #

Hi,
how can i make future prediction if i have used the following function to make prediction :
for timepoint in range(len(TestData)):
ActualValue = TestData[timepoint]
#forcast value
Prediction = StartARIMAForecasting(Actual, 1,1,1)
print('Actual=%f, Predicted=%f' % (ActualValue, Prediction))
#add it in the list
Predictions.append(Prediction)
Actual.append(ActualValue)
and thanks

Jason Brownlee April 30, 2019 at 2:25 pm #

REPLY

You can use model.predict() or model.forecast() as specified in the post.

ayushi saxena May 16, 2019 at 6:29 pm #

REPLY

hi,
please tell why it is not working correctly:
a=[1,2,3,4,1,2,3,4]
da = difference(a)
X=a
forecast=da
forecast
[1, 1, 1, -3, 1, 1, 1]

days_in_year=4
history = [x for x in X]
day = 1
for yhat in forecast:
inverted = inverse_difference(history, yhat, days_in
print('Day %d: %f' % (day, inverted))
history.append(inverted)
day += 1
history
Day 1: 2.000000
Day 2: 3.000000
Day 3: 4.000000
Day 4: 1.000000
Day 5: 3.000000
Day 6: 4.000000
Day 7: 5.000000
why day5 is incorrect?

Jason Brownlee May 17, 2019 at 5:51 am #

REPLY

I have some suggestions here:
https://machinelearningmastery.com/faq/single-faq/can-you-read-review-or-debug-my-code

mee May 18, 2019 at 10:53 pm #

REPLY

Hi,
how can i calculate RMSE and other indictors of performance ?
thank you

Jason Brownlee May 19, 2019 at 8:02 am #

REPLY

You can use the sklearn metrics to calculate the error between an array of predictions and an array of expected values:
https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics

Shi May 22, 2019 at 8:23 pm #

REPLY

Hi Jason,
Your blog are very helpful. I applied ARIMA by setting the train and test data by ratios (like, 90:10, 80:20, 70:30..) for prediction. i thought RMSE value reduces as the train data increases. but i got the below answer when i predicted for 5 years of data.
Ratio MSE RMSE
90-10 116.18 10.779
80-20 124.336 11.151
70-30 124.004 11.136
60-40 126.268 11.237
50-50 127.793 11.305
40-60 137.029 11.706
30-70 133.29 11.545

So, now i got confused. The RMSE has to reduce a
can you tell me what are the possible reasons for v

thank you

Jason Brownlee May 23, 2019 at 6:02 a

Variation in reported error scores is ba
interval being predicted.

It is a good idea to summarise the performance
large interval.

baktr_ May 23, 2019 at 2:34 am #

REPLY

hi,thanks for your blog but i need support. when i run code :

```
def difference(dataset, interval=1):
diff =list()
for i in range(interval, len(dataset)):
value = dataset[i]-dataset[i-interval]
diff.append(value)
return numpy.array(diff)

df = pd.read_csv('dataset.csv',header=None)
X = df.values
day_in_year = 365
differenced = difference(X,day_in_year)

model =ARIMA(differenced,order=(7,0,1))
model_fit=model.fit(disp=0)
print(model_fit.summary())
```

and

TypeError Traceback (most recent call last)
in
9 X = df.values
10 day_in_year = 365
—> 11 differenced = difference(X,day_in_year)
12
13 model =ARIMA(differenced,order=(7,0,1))

in difference(dataset, interval)
2 diff =list()
3 for i in range(interval, len(dataset)):
—-> 4 value = dataset[i]-dataset[i-interval]
5 diff.append(value)
6 return numpy.array(diff)

TypeError: unsupported operand type(s) for -: 'str'

i don't know what it's mean. i run it in python3, can

Jason Brownlee May 23, 2019 at 6:05 a

Sorry to hear that, I have some sugge
https://machinelearningmastery.com/faq/single
me

Nora.M July 14, 2019 at 11:05 am #

REPLY

please i want to apply this code to time series data , but i want to make sliding window that take the first five values and predict the six and make sliding to the next ,what should i change to build this model

Jason Brownlee July 15, 2019 at 8:15 am #

REPLY

You can change the order to be something like (5,1,0) and use the forecast() function with the number of steps set to 6.

Jia Ying July 29, 2019 at 7:33 pm #

REPLY

Hi Jason!

I would like to make a out of sample prediction of the data. However, from what I have seen from your tutorial as well as other posts online, most of the prediction seemed more like a validation of the data that they are already have.

E.g. I have the annual population data from 1950-2019
I split the data into the train data(1950 -1998) and the test data (1998 onwards to 2019).

Of course I start off with creating my model using the sample data, then doing a validation using the test data. But how do I predict the annual population beyond 2019?

Thank you so much!

Jason Brownlee July 30, 2019 at 6:08 am #

REPLY

Good question.

Fit your model on all available data to create a
forecast() or predict() for the interval you wish t

Jia Ying July 30, 2019 at 7:06 pm #

Thank you so much for your prom

Another question. I am actually using auto_
how the predict function in auto_arima wor
end parameters. The parameters are (from
how is the algorithm supposed to know if y
prediction?

This was how I used it in my code.
test is the test data whereas train is the training data

newforecast is basically the predicted value for the test data. However, I would like to do a out-sample prediction instead.

```
import pmdarima as pm

for ctry in seadict.keys():
dataa = seadict[ctry]
slicing = int(len(dataa)*0.7)
train = dataa[0:slicing]
test=dataa[slicing:len(dataa)]
mod = pm.auto_arima(train, error_action='ignore', suppress_warnings = True)
mod.fit(train)
forecast = mod.predict(n_periods=len(test))
newforecast = pd.Series(forecast, index=test.index)
```

Jason Brownlee July 31, 2019 at 6:48 am #

REPLY

I am not familiar with that library, sorry.

I have my own implementation here that might help:
https://machinelearningmastery.com/h

time-series-forecasting-in-python/

Arij August 21, 2019 at 6:20 pm #

REPLY

Hi how can i install the dataset?
the link just shows the data on webpage

Jason Brownlee August 22, 2019 at 6:24 am #

REPLY

Download the dataset as a .csv file in

Mark Lavin October 24, 2019 at 2:05 am #

I have a time series that's on a monthly ca
the values using an ARIMA model, but I keep gettin
specify one of the missing dates using "start=missi
using "exog = [missing_date]" there is no error bu
gaps) that was used to fit the ARIMA model. I'm sta
"interpolate" using ARIMA; is that correct?

Jason Brownlee October 24, 2019 at 5:41 am #

REPLY

Filling in missing values with ARIMA is hard, you may have to fit a model that ends prior to each gap and then predict the gap.

Also try the forecast() function, it is much easier.

HARIHARAN K November 16, 2019 at 3:41 am #

REPLY

difference function is doing the difference between current and previous day value not the previous year value. You are describing it as year in the post. Hope i'm correct

Jason Brownlee November 16, 2019 at 7:27 am #

REPLY

Look at how we call the function and pass in 365.

Sagar December 20, 2019 at 7:03 am #

REPLY

Start Machine Learning

Hi,

Thanks for your tutorials. They are amazing.

I had to make the following changes to make the code work. Notice that had to use index [1] in line 5 and the last line. Ami I doing some thing wrong?

Appreciate if you can point out my error. I am using Anaconda 3.5

```
# create a differenced series
def difference(dataset, interval=1):
diff = list()
for i in range(interval, len(dataset)):
value = dataset[i][1] – dataset[i – interval][1]
diff.append(value)
return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
return yhat + history[-interval][1]
```


Jason Brownlee December 20, 2019 at 1

You're welcome, thanks for your kind

Sorry to hear that, perhaps confirm that you ar
code and data exactly.

Also this might help:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

REPLY

sagar December 20, 2019 at 1:37 pm #

I think I know where the problem is. It is in the read statement. I am trying figure out a way to read correctly

REPLY

Jason Brownlee December 21, 2019 at 7:05 am #

Great!

REPLY

Ébe January 7, 2020 at 6:17 pm #

A nice yet concise tutorial, Dr. Jason Brownlee!

I have a basic question I still couldn't get the answe
arima.model.ARIMAResults.forecast()?

The output according to its docs is “Array of out of sample forecasts. A (steps x k_endog) array.” I’m sure endog means the input array used as history for training, and steps is the specified integer parameter. I’m not sure what k_endog means.

Could you please let us know?

Thanks

Jason Brownlee January 8, 2020 at 8:20 am #

REPLY

Thanks!

I believe the forecasted interval and the predict

Dulanja Gunawardena January 21, 2020 at

When the code is compiled, this error sho

File “C:/Users/D.T/.spyder-py3/untitled1.py”, line 9,
value = dataset[i] – dataset[i – interval]

TypeError: unsupported operand type(s) for -: ‘str’

Please Help !!

Jason Brownlee January 22, 2020 at 6:18 am #

REPLY

Sorry to hear that, this might help:
https://machinelearningmastery.com/faq/single-faq/why-does-the-code-in-the-tutorial-not-work-for-me

Harshit Musugu February 7, 2020 at 5:12 am #

REPLY

The end argument could not be matched to a location related to the index of the data.’

This is what I am getting when I use :

pred = res.predict(start = ‘2014-11-05’, end = ‘2019-02-01’)

How to do out of forecast predictions when we have date as our index

Jason Brownlee February 7, 2020 at 8:25 am #

REPLY

I don’t have an example off the cuff, sorry.

Abhay Saini February 13, 2020 at 8:47 pm #

REPLY

Hi Jason,

Firstly thanks a ton for useful blogs!
I had a doubt in this one:- https://machinelearningmastery.com/make-sample-forecasts-arima-python/

You have used predict function to make out of sample forecasts.
However when i tried it ;-
1) I was only able to run the predict function on start and end indexes as numbers and not dates
2) If i give a number below len(series) (in our case differenced), will i get a forecast of a subset of the training data itself? Meaning, i can easily compare actual/predicted like we do in linear regression?
Because everywhere, you have discussed about o
Thanks,
Abhay

Jason Brownlee February 14, 2020 at 6:

I don't have examples of forecasting v

You can predict the train set, but it is better to u
https://machinelearningmastery.com/backtest-r

rodney February 21, 2020 at 9:03 pm #

i only have daily data for four months in one year and i want to forecast to sales for the coming years. how can i do it. because i see from the difference that you comparing with data of the same period from the previous year which i dont have. How can i forecast with my limited data.

Jason Brownlee February 22, 2020 at 6:24 am #

REPLY

Fit the model on available data and call model.predict().

Perhaps I don't understand the problem you're having exactly?

Mukesh February 25, 2020 at 2:26 am #

REPLY

Hello Jason I'm using python 3.7.4
but still there is problem with
TypeError Traceback (most recent call last)
in
16 X = series.values
17 days_in_year = 365
—> 18 differenced = difference(X, days_in_year)

```
19 # fit model
20 model = ARIMA(differenced, order=(7,0,1))

in difference(dataset, interval)
7 diff = list()
8 for i in range(interval, len(dataset)):
—-> 9 value = dataset[i] – dataset[i – interval]
10 diff.append(value)
11 return numpy.array(diff)
```

TypeError: unsupported operand type(s) for -: ‘str’ and ‘str’

Your tutorials help me alot and I started my machine learning journey by following youre website and email newsletter.
please help me with this issue I tried all the ways

Jason Brownlee February 25, 2020 at 7:

Sorry to hear that, this will help:
https://machinelearningmastery.com/faq/single
me

manjunath March 5, 2020 at 2:09 am #

can we have same functions in RNN ?
Please share if you have post

Jason Brownlee March 5, 2020 at 6:39 am #

REPLY

Yes, you can start here:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome!

My name is *Jason Brownlee* PhD, and I

Read more

Never miss a tutorial:

Picked for you:

ARIMA Rolling How to Create an ARIMA Model for Time Series

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

Time Series Forecas Time Series Forecasting as Supervised Learning

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

How To Backtest How To Backtest Machine Learning Models for Time Series Forecasting

Loving the Tutorials?

The Time Series with Python EBook
is where I keep the ***Really Good*** stuff.

SEE WHAT'S INSIDE

Start Machine Learning