

[Start Here](#)[Blog](#)[Books](#)[About](#)[Contact](#)

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

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



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.

Let's get started.

[Get Your Start in Machine Learning](#)



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

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Tutorial Overview

This tutorial is broken down into the following 5 steps:

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

Stop learning Time Series Forecasting the

Get Your Start in Machine Learning

Take my free 7-day email course and discover data prep, modeling and more (with sample code).

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

Start Your FREE Mini-Course Now!

1. Minimum Daily Temperatures Dataset

This dataset describes the minimum daily temperatures over 10 years (1981-1990) in the city of Melbourne, Australia.

The units are in degrees Celsius and there are 3,650 observations. The source of the data is credited to [Data Market](#).

[Learn more about the dataset on Data Market.](#)

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

Note: The downloaded file contains some question mark (“?”) characters that must be removed before using the file in a Python script. Use a text editor and remove the “?” characters. Also, remove any footer information in the file.

The example below loads the dataset as a Pandas Series.

```
1 # line plot of time series
2 from pandas import Series
3 from matplotlib import pyplot
4 # load dataset
5 series = Series.from_csv('daily-minimum-temperatures.csv', header=0)
6 # display first few rows
7 print(series.head(20))
8 # line plot of dataset
9 series.plot()
10 pyplot.show()
```

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

1 Date

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

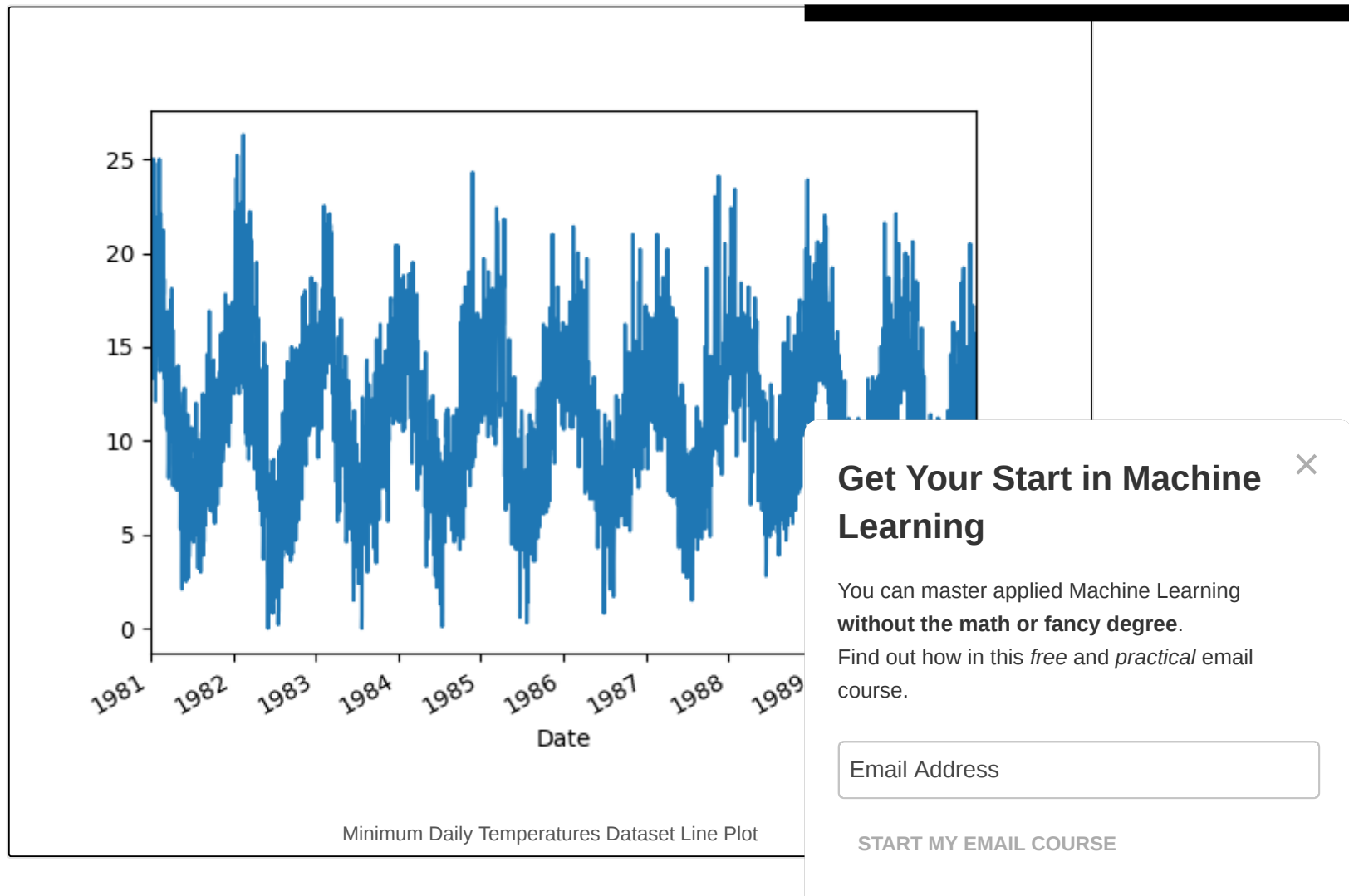
2	1981-01-01	20.7
3	1981-01-02	17.9
4	1981-01-03	18.8
5	1981-01-04	14.6
6	1981-01-05	15.8
7	1981-01-06	15.8
8	1981-01-07	15.8
9	1981-01-08	17.4
10	1981-01-09	21.8
11	1981-01-10	20.0
12	1981-01-11	16.2
13	1981-01-12	13.3
14	1981-01-13	16.7
15	1981-01-14	21.5
16	1981-01-15	25.0
17	1981-01-16	20.7
18	1981-01-17	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.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



2. Split Dataset

We can split the dataset into two parts.

The first part is the training dataset that we will use to prepare an ARIMA model. The second part is the test dataset that we will pretend is not available. It is these time steps that we will treat as out of sample.

The dataset contains data from January 1st 1981 to December 31st 1990.

Get Your Start in Machine Learning

We will hold back the last 7 days of the dataset from December 1990 as the test dataset and treat those 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.

```
1 # split the dataset
2 from pandas import Series
3 series = Series.from_csv('daily-minimum-temperatures.csv', header=0)
4 split_point = len(series) - 7
5 dataset, validation = series[0:split_point], series[split_point:]
6 print('Dataset %d, Validation %d' % (len(dataset), len(validation)))
7 dataset.to_csv('dataset.csv')
8 validation.to_csv('validation.csv')
```

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 data

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.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```

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

```

We can invert this operation by adding the value of the observation one year ago. We will need to do this to any forecasts made by a model trained on the seasonally adjusted data.

```

1 # invert differenced value
2 def inverse_difference(history, yhat, interval=1):
3     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 parameters, I will choose a simple ARIMA(7,0,7) configuration.

We can put all of this together as follows:

```

1 from pandas import Series
2 from statsmodels.tsa.arima_model import ARIMA
3 import numpy
4
5 # create a differenced series
6 def difference(dataset, interval=1):
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)
12
13 # load dataset
14 series = Series.from_csv('dataset.csv', header=None)
15 # seasonal difference
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))
21 model_fit = model.fit(dis=0)
22 # print summary of fit model

```

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
23 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.

```
1
2
3 ARMA Model Results
4 =====
5 Dep. Variable:          y      No. Observations:      3278
6 Model:                ARMA(7, 1)  Log Likelihood      -8673.748
7 Method:                css-mle   S.D. of innovations      3.411
8 Date:                  Mon, 20 Feb 2017  AIC              17367.497
9 Time:                  10:28:38    BIC              17428.447
10 Sample:                0      HQIC              17389.322
11
12 =====
13
14      coef      std err          z      P>|z|      [0.025      0.975]
15 -----
16 const          0.0132      0.132      0.100      0.921      -0.246      0.273
17 ar.L1.y         1.1424      0.287      3.976      0.000      0.579      1.706
18 ar.L2.y        -0.4346      0.154     -2.829      0.005     -0.736     -0.133
19 ar.L3.y         0.0961      0.042      2.289      0.022      0.014      0.178
20 ar.L4.y         0.0125      0.029      0.434      0.664     -0.044      0.069
21 ar.L5.y        -0.0101      0.029     -0.343      0.732     -0.068      0.047
22 ar.L6.y         0.0119      0.027      0.448      0.654     -0.040      0.064
23 ar.L7.y         0.0089      0.024      0.368      0.713     -0.038      0.056
24 ma.L1.y        -0.6157      0.287     -2.146      0.032     -1.178     -0.053
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

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.

Get Your Start in Machine Learning

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:

```
1 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.

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

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

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

The complete example is listed below:

```
1 from pandas import Series
2 from statsmodels.tsa.arima_model import ARIMA
3 import numpy
4
5 # create a differenced series
6 def difference(dataset, interval=1):
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)
12
13 # invert differenced value
14 def inverse_difference(history, yhat, interval=1):
15     return yhat + history[-interval]
16
17 # load dataset
```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```

18 series = Series.from_csv('dataset.csv', header=None)
19 # seasonal difference
20 X = series.values
21 days_in_year = 365
22 differenced = difference(X, days_in_year)
23 # fit model
24 model = ARIMA(differenced, order=(7,0,1))
25 model_fit = model.fit(dispatch=0)
26 # one-step out-of sample forecast
27 forecast = model_fit.forecast()[0]
28 # invert the differenced forecast to something usable
29 forecast = inverse_difference(X, forecast, days_in_year)
30 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.

```
1 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 predict function requires a start and an end to be specified, these can be the indexes of the time used to fit the model, for example:

```

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

```

The start and end can also be a datetime string or a “datetime” type; for example:

```

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

```

and

```

1 from pandas import datetime
2 start_index = datetime(1990, 12, 25)

```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```

3 end_index = datetime(1990, 12, 26)
4 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:

```

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

```

Perhaps you will have more luck; for now, I am sticking with the time step indexes.

The complete example is listed below:

```

1 from pandas import Series
2 from statsmodels.tsa.arima_model import ARIMA
3 import numpy
4 from pandas import datetime
5
6 # create a differenced series
7 def difference(dataset, interval=1):
8     diff = list()
9     for i in range(interval, len(dataset)):
10         value = dataset[i] - dataset[i - interval]
11         diff.append(value)
12     return numpy.array(diff)
13
14 # invert differenced value
15 def inverse_difference(history, yhat, interval=1):
16     return yhat + history[-interval]
17
18 # load dataset
19 series = Series.from_csv('dataset.csv', header=None)
20 # seasonal difference
21 X = series.values
22 days_in_year = 365
23 differenced = difference(X, days_in_year)
24 # fit model
25 model = ARIMA(differenced, order=(7,0,1))
26 model_fit = model.fit(dis=0)
27 # one-step out of sample forecast
28 start_index = len(differenced)
29 end_index = len(differenced)
30 forecast = model_fit.predict(start=start_index, end=end_index)
31 # invert the differenced forecast to something usable
32 forecast = inverse_difference(X, forecast, days_in_year)
33 print('Forecast: %f' % forecast)

```

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

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

```
1 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 Forecast

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.

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 7-step forecast.

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

We can then invert each forecasted time step, one at a time and print the values. Note that to invert the forecast value for $t+1$. Here, we add them to the end of a list called `history` for use when calling `invert`.

```
1 # invert the differenced forecast to something usable
2 history = [x for x in X]
3 day = 1
4 for yhat in forecast:
5     inverted = inverse_difference(history, yhat, days_in_year)
6     print('Day %d: %f' % (day, inverted))
7     history.append(inverted)
8     day += 1
```

The complete example is listed below:

```
1 from pandas import Series
```

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```

2 from statsmodels.tsa.arima_model import ARIMA
3 import numpy
4
5 # create a differenced series
6 def difference(dataset, interval=1):
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)
12
13 # invert differenced value
14 def inverse_difference(history, yhat, interval=1):
15     return yhat + history[-interval]
16
17 # load dataset
18 series = Series.from_csv('dataset.csv', header=None)
19 # seasonal difference
20 X = series.values
21 days_in_year = 365
22 differenced = difference(X, days_in_year)
23 # fit model
24 model = ARIMA(differenced, order=(7,0,1))
25 model_fit = model.fit(dis=0)
26 # multi-step out-of-sample forecast
27 forecast = model_fit.forecast(steps=7)[0]
28 # invert the differenced forecast to something usable
29 history = [x for x in X]
30 day = 1
31 for yhat in forecast:
32     inverted = inverse_difference(history, yhat, days_in_year)
33     print('Day %d: %f' % (day, inverted))
34     history.append(inverted)
35     day += 1

```

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

```

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

```

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

Predict Function

The `predict()` function can also forecast the next 7 out-of-sample time steps.

Using time step indexes, we can specify the end index as 6 more time steps in the future; for example:

```
1 # multi-step out-of-sample forecast
2 start_index = len(differenced)
3 end_index = start_index + 6
4 forecast = model_fit.predict(start=start_index, end=end_index)
```

The complete example is listed below.

```
1 from pandas import Series
2 from statsmodels.tsa.arima_model import ARIMA
3 import numpy
4
5 # create a differenced series
6 def difference(dataset, interval=1):
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)
12
13 # invert differenced value
14 def inverse_difference(history, yhat, interval=1):
15     return yhat + history[-interval]
16
17 # load dataset
18 series = Series.from_csv('dataset.csv', header=None)
19 # seasonal difference
20 X = series.values
21 days_in_year = 365
22 differenced = difference(X, days_in_year)
23 # fit model
24 model = ARIMA(differenced, order=(7,0,1))
25 model_fit = model.fit(dispatch=0)
26 # multi-step out-of-sample forecast
27 start_index = len(differenced)
28 end_index = start_index + 6
29 forecast = model_fit.predict(start=start_index, end=end_index)
30 # invert the differenced forecast to something usable
31 history = [x for x in X]
32 day = 1
```

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
33 for yhat in forecast:
34     inverted = inverse_difference(history, yhat, days_in_year)
35     print('Day %d: %f' % (day, inverted))
36     history.append(inverted)
37     day += 1
```

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

```
1 Day 1: 14.861669
2 Day 2: 15.628784
3 Day 3: 13.331349
4 Day 4: 11.722413
5 Day 5: 10.421523
6 Day 6: 14.415549
7 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 forecast.
- How to make a 7-day multi-step out-of-sample forecast.
- How to use both the `forecast()` and `predict()` functions when forecasting.

Do you have any questions about out-of-sample forecasts, or about this post? Ask your questions in

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

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](#)

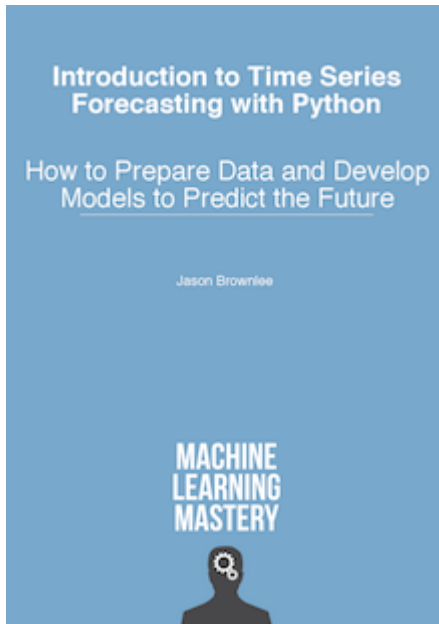
Get Your Start in Machine Learning

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.

[Click to learn more.](#)



About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional developer, and entrepreneur. He is passionate about helping developers get started and get good at applied machine learning. [Learn more.](#)

[View all posts by Jason Brownlee](#) →

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

[← Time Series Forecasting with Python 7-Day Mini-Course](#)

[Sensitivity Analysis of History Size to Forecast Skill with ARIMA in Python >](#)

[Get Your Start in Machine Learning](#)

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



Steve March 24, 2017 at 10:44 pm #

REPLY ↩

Your tutorials are the most helpful machine learning resources I have found on the Internet and have been hugely helpful in work and personal side projects. I don't know if you take requests but I'd love to see a series of posts on recommender systems one of these days!



Jason Brownlee March 25, 2017 at 7:36 am #

Thanks Steve, and great suggestion! Thanks.



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? I say I have parameters that were estimated using a dataset that I no longer have but I still want to produce forecasts.

Thanks

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



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/>

Get Your Start in Machine Learning



masum May 11, 2017 at 8:32 pm #

REPLY ↩

sir,

would it be possible to do the same using LSTM RNN ?

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

Thanking you



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

REPLY ↩

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-network/>

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning



masum May 12, 2017 at 8:29 pm #

I tried to run the above example without any seasonal difference with given below code.

```
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-temperatures.csv', header=0)
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(dispatch=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



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 #

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 pm #

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

See this post:

<http://machinelearningmastery.com/seasonal-persistence-forecasting-python/>

And this post:

<http://machinelearningmastery.com/time-series-seasonality-with-python/>

Please use the search feature of the blog.



Hans June 15, 2017 at 11:27 am #

REPLY ↩

If I pretend data in test-partition is not given, does this tutorial do the same except of the seasonal cleaning?

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



Hans June 15, 2017 at 11:29 am #

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



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

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 #

REPLY ↩

I did so several times.

How can I obtain a train RMSE from the model?

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning



Jason Brownlee June 17, 2017 at 7:20 am #

REPLY ↩

See this post on how to estimate the skill of a model prior to using it to make out of sample predictions:
<http://machinelearningmastery.com/backtest-machine-learning-models-time-series-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-

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree**.
Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



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.

Get Your Start in Machine Learning



Hans June 16, 2017 at 3:06 pm #

REPLY ↩

Is this the line where the training happens?

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



Jason Brownlee June 17, 2017 at 7:22 am #

REPLY ↩

No here:

```
1 model_fit = model.fit(dispatch=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.



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?



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

REPLY ↩

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.

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**
Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

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 ARIMA 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 forecasted 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 #

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/>

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



AMU June 23, 2017 at 5:33 am #

Thank you Jason for this wonderful post... It is very detailed and easy to understand..

Do you also have something similar for LSTM Neural Network algorithm as well? something like – How to make out of sample forecasts with LSTM in Python.

If not, will you write one blog like this with detail explanation? I am sure there are lot of people have the same question.



Jason Brownlee June 23, 2017 at 6:45 am #

REPLY ↩

Almost every post I have on LSTMs shows how to make out of sample forecasts. The code is provided in the next forecast validation

Get Your Start in Machine Learning



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 #

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 apples. This may be less of a concern for an R^2 .

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



Vishanth July 19, 2017 at 6:56 am #

REPLY ↩

Seriously amazing. Thanks a lot professor

Get Your Start in Machine Learning



Jason Brownlee July 19, 2017 at 8:30 am #

REPLY ↩

Thanks. Also, I'm not a professor.



Kirui July 20, 2017 at 5:15 pm #

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.

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



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

Perhaps check that you have loaded your data correct (as real values) and that you have c
space.

REPLY ↩



Antoine August 23, 2017 at 1:00 am #

Hi Jason,

Thanks for this detailed 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 ?

Get Your Start in Machine Learning

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 !



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

REPLY ↩

Yes, you can use a grid search:

<https://machinelearningmastery.com/grid-search-arima-hyperparameters-with-python/>



Bob October 6, 2017 at 11:53 pm #

Ciao Jason,

Thanks for this tutorial and all the time series related ones. There is always a sense of order in how you I'm by the way still confused about something which is probably more conceptual about ARIMA.

The ARIMA parameters specify the lag which it uses to forecast.

In your case you used $p=7$ for example so that you would take into consideration the previous week.

A first silly question is why do I need to fit an entire year of data if I'm only looking at my window/lags ?

The second question is that fitting my model I get an error which is really minimal even if I use a short training first point.

What am I missing?

Thanks

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



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

REPLY ↩

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.

Get Your Start in Machine Learning

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

Welcome to Machine Learning Mastery



Hi, I'm Dr. Jason Brownlee.

My goal is to make practitioners like YOU awesome at applied machine learning.

[Read More](#)

Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

Get Good at Time Series Forecasting

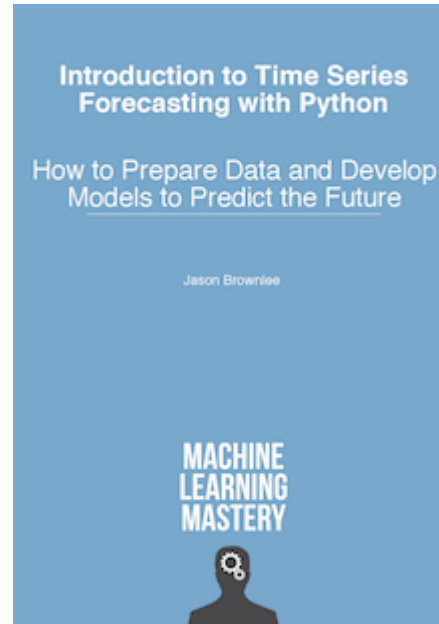
Get Your Start in Machine Learning

Need visualizations and forecast models?

Looking for step-by-step tutorials?

Want end-to-end projects?

Get Started with Time Series Forecasting in Python!



Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

POPULAR



Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras

JULY 21, 2016



Your First Machine Learning Project in Python Step-By-Step

JUNE 10, 2016



Develop Your First Neural Network in Python With Keras Step-By-Step

MAY 24, 2016

Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras

Get Your Start in Machine Learning



JULY 26, 2016



How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda

MARCH 13, 2017



Time Series Forecasting with the Long Short-Term Memory Network in Python

APRIL 7, 2017



Multi-Class Classification Tutorial with the Keras Deep Learning Library

JUNE 2, 2016



Regression Tutorial with the Keras Deep Learning Library in Python

JUNE 9, 2016



Multivariate Time Series Forecasting with LSTMs in Keras

AUGUST 14, 2017



How to Implement the Backpropagation Algorithm From Scratch In Python

NOVEMBER 7, 2016

Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

© 2017 Machine Learning Mastery. All Rights Reserved.

[Privacy](#) | [Contact](#) | [About](#)

Get Your Start in Machine Learning