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

by **Jason Brownlee** on August 6, 2018 in **Time Series**

Machine learning methods can be used for classification and forecasting on time series problems.

Before exploring machine learning methods for time series, it is a good idea to ensure you have exhausted classical linear time series forecasting methods. Classical time series forecasting methods may be focused on linear relationships, nevertheless, they are sophisticated and perform well on a wide range of problems, assuming that your data is suitably prepared and the method is well configured.

In this post, will you will discover a suite of classical methods for time series forecasting that you can test on your forecasting problem prior to exploring to machine learning methods.

The post is structured as a cheat sheet to give you just enough information on each method to get started with a working code example and where to look to get more information on the method.

All code examples are in Python and use the Statsmodels library. The APIs for this library can be tricky for beginners (trust me!), so having a working code example as a starting point will greatly accelerate your progress.

This is a large post; you may want to bookmark it.

Let’s get started.

11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)
Photo by Ron Reiring, some rights reserved.

Overview

This cheat sheet demonstrates 11 different classical time series forecasting methods; they are:

1. Autoregression (AR)
2. Moving Average (MA)
3. Autoregressive Moving Average (ARMA)
4. Autoregressive Integrated Moving Average (ARIMA)
5. Seasonal Autoregressive Integrated Moving-Average (SARIMA)
6. Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX)
7. Vector Autoregression (VAR)
8. Vector Autoregression Moving-Average (VARMA)
9. Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX)
10. Simple Exponential Smoothing (SES)
11. Holt Winter’s Exponential Smoothing (HWES)

Did I miss your favorite classical time series forecasting method?
Let me know in the comments below.

Each method is presented in a consistent manner.

This includes:

- **Description**. A short and precise description of the technique.

- **Python Code**. A short working example of fitting the model and making a prediction in Python.
- **More Information**. References for the API and the algorithm.

Each code example is demonstrated on a simple contrived dataset that may or may not be appropriate for the method. Replace the contrived dataset with your data in order to test the method.

Remember: each method will require tuning to your specific problem. In many cases, I have examples of how to configure and even grid search parameters on the blog already, try the search function.

If you find this cheat sheet useful, please let me know in the comments below.

Autoregression (AR)

The autoregression (AR) method models the next step in the sequence as a linear function of the observations at prior time steps.

The notation for the model involves specifying the order of the model p as a parameter to the AR function, e.g. AR(p). For example, AR(1) is a first-order autoregression model.

The method is suitable for univariate time series without trend and seasonal components.

Python Code

```python
# AR example
from statsmodels.tsa.ar_model import AR
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = AR(data)
model_fit = model.fit()
# make prediction
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

More Information

- statsmodels.tsa.ar_model.AR API
- statsmodels.tsa.ar_model.ARResults API
- Autoregressive model on Wikipedia

Moving Average (MA)

The moving average (MA) method models the next step in the sequence as a linear function of the residual errors from a mean process at prior time steps.

A moving average model is different from calculating the moving average of the time series.

The notation for the model involves specifying the order of the model q as a parameter to the MA function, e.g. MA(q). For example, MA(1) is a first-order moving average model.

The method is suitable for univariate time series without trend and seasonal components.

Python Code

We can use the ARMA class to create an MA model and setting a zeroth-order AR model. We must specify the order of the MA model in the order argument.

```
# MA example
from statsmodels.tsa.arima_model import ARMA
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = ARMA(data, order=(0, 1))
model_fit = model.fit(disp=False)
# make prediction
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

More Information

- statsmodels.tsa.arima_model.ARMA API
- statsmodels.tsa.arima_model.ARMAResults API
- Moving-average model on Wikipedia

Autoregressive Moving Average (ARMA)

The Autoregressive Moving Average (ARMA) method models the next step in the sequence as a linear function of the observations and resiudal errors at prior time steps.

It combines both Autoregression (AR) and Moving Average (MA) models.

The notation for the model involves specifying the order for the AR(p) and MA(q) models as parameters to an ARMA function, e.g. ARMA(p, q). An ARIMA model can be used to develop AR or MA models.

The method is suitable for univariate time series without trend and seasonal components.

Python Code

```
# ARMA example
from statsmodels.tsa.arima_model import ARMA
from random import random
# contrived dataset
data = [random() for x in range(1, 100)]
# fit model
model = ARMA(data, order=(2, 1))
model_fit = model.fit(disp=False)
# make prediction
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

More Information

- statsmodels.tsa.arima_model.ARMA API
- statsmodels.tsa.arima_model.ARMAResults API
- Autoregressive–moving-average model on Wikipedia

Autoregressive Integrated Moving Average (ARIMA)

The Autoregressive Integrated Moving Average (ARIMA) method models the next step in the sequence as a linear function of the differenced observations and residual errors at prior time steps.

It combines both Autoregression (AR) and Moving Average (MA) models as well as a differencing pre-processing step of the sequence to make the sequence stationary, called integration (I).

The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function, e.g. ARIMA(p, d, q). An ARIMA model can also be used to develop AR, MA, and ARMA models.

The method is suitable for univariate time series with trend and without seasonal components.

Python Code

```
# ARIMA example
from statsmodels.tsa.arima_model import ARIMA
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = ARIMA(data, order=(1, 1, 1))
model_fit = model.fit(disp=False)
# make prediction
yhat = model_fit.predict(len(data), len(data), typ='levels')
print(yhat)
```

More Information

- statsmodels.tsa.arima_model.ARIMA API
- statsmodels.tsa.arima_model.ARIMAResults API
- Autoregressive integrated moving average on Wikipedia

Seasonal Autoregressive Integrated Moving-Average (SARIMA)

The Seasonal Autoregressive Integrated Moving Average (SARIMA) method models the next step in the sequence as a linear function of the differenced observations, errors, differenced seasonal observations, and seasonal errors at prior time steps.

It combines the ARIMA model with the ability to perform the same autoregression, differencing, and moving average modeling at the seasonal level.

The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function and AR(P), I(D), MA(Q) and m parameters at the seasonal level, e.g. SARIMA(p, d, q)(P, D, Q)m where “m” is the number of time steps in each season (the seasonal period). A SARIMA model can be used to develop AR, MA, ARMA and ARIMA models.

The method is suitable for univariate time series with trend and/or seasonal components.

Python Code

```
# SARIMA example
from statsmodels.tsa.statespace.sarimax import SARIMAX
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = SARIMAX(data, order=(1, 1, 1), seasonal_order=(1, 1, 1, 1))
model_fit = model.fit(disp=False)
# make prediction
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

More Information

- statsmodels.tsa.statespace.sarimax.SARIMAX API
- statsmodels.tsa.statespace.sarimax.SARIMAXResults API
- Autoregressive integrated moving average on Wikipedia

Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX)

The Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX) is an extension of the SARIMA model that also includes the modeling of exogenous variables.

Exogenous variables are also called covariates and can be thought of as parallel input sequences that have observations at the same time steps as the original series. The primary series may be referred to as endogenous data to contrast it from the exogenous sequence(s). The observations for exogenous variables are included in the model directly at each time step and are not modeled in the same way as the primary endogenous sequence (e.g. as an AR, MA, etc. process).

The SARIMAX method can also be used to model the subsumed models with exogenous variables, such as ARX, MAX, ARMAX, and ARIMAX.

The method is suitable for univariate time series with trend and/or seasonal components and exogenous variables.

Python Code

```python
# SARIMAX example
from statsmodels.tsa.statespace.sarimax import SARIMAX
from random import random
# contrived dataset
data1 = [x + random() for x in range(1, 100)]
data2 = [x + random() for x in range(101, 200)]
# fit model
model = SARIMAX(data1, exog=data2, order=(1, 1, 1), seasonal_order=(0, 0, 0, 0))
model_fit = model.fit(disp=False)
# make prediction
exog2 = [200 + random()]
yhat = model_fit.predict(len(data1), len(data1), exog=[exog2])
print(yhat)
```

More Information

- statsmodels.tsa.statespace.sarimax.SARIMAX API
- statsmodels.tsa.statespace.sarimax.SARIMAXResults API
- Autoregressive integrated moving average on Wikipedia

Vector Autoregression (VAR)

The Vector Autoregression (VAR) method models the next step in each time series using an AR model. It is the generalization of AR to multiple parallel time series, e.g. multivariate time series.

The notation for the model involves specifying the order for the AR(p) model as parameters to a VAR function, e.g. VAR(p).

The method is suitable for multivariate time series without trend and seasonal components.

Python Code

```python
# VAR example
from statsmodels.tsa.vector_ar.var_model import VAR
from random import random
# contrived dataset with dependency
data = list()
for i in range(100):
v1 = i + random()
v2 = v1 + random()
row = [v1, v2]
data.append(row)
# fit model
model = VAR(data)
model_fit = model.fit()
# make prediction
yhat = model_fit.forecast(model_fit.y, steps=1)
print(yhat)
```

More Information

- statsmodels.tsa.vector_ar.var_model.VAR API

- statsmodels.tsa.vector_ar.var_model.VARResults API
- Vector autoregression on Wikipedia

Vector Autoregression Moving-Average (VARMA)

The Vector Autoregression Moving-Average (VARMA) method models the next step in each time series using an ARMA model. It is the generalization of ARMA to multiple parallel time series, e.g. multivariate time series.

The notation for the model involves specifying the order for the AR(p) and MA(q) models as parameters to a VARMA function, e.g. VARMA(p, q). A VARMA model can also be used to develop VAR or VMA models.

The method is suitable for multivariate time series without trend and seasonal components.

Python Code

```python
# VARMA example
from statsmodels.tsa.statespace.varmax import VARMAX
from random import random
# contrived dataset with dependency
data = list()
for i in range(100):
v1 = random()
v2 = v1 + random()
row = [v1, v2]
data.append(row)
# fit model
model = VARMAX(data, order=(1, 1))
model_fit = model.fit(disp=False)
# make prediction
yhat = model_fit.forecast()
print(yhat)
```

More Information

- statsmodels.tsa.statespace.varmax.VARMAX API
- statsmodels.tsa.statespace.varmax.VARMAXResults
- Vector autoregression on Wikipedia

Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX)

The Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX) is an extension of the VARMA model that also includes the modeling of exogenous variables. It is a multivariate version of the ARMAX method.

Exogenous variables are also called covariates and can be thought of as parallel input sequences that have observations at the same time steps as the original series. The primary series(es) are referred to as endogenous data to contrast it from the exogenous sequence(s). The observations for exogenous variables are included in the model directly at each time step and are not modeled in the same way as the primary endogenous sequence (e.g. as an AR, MA, etc. process).

The VARMAX method can also be used to model the subsumed models with exogenous variables, such as VARX and VMAX.

The method is suitable for multivariate time series without trend and seasonal components with exogenous variables.

Python Code

```
# VARMAX example
from statsmodels.tsa.statespace.varmax import VARMAX
from random import random
# contrived dataset with dependency
data = list()
for i in range(100):
v1 = random()
v2 = v1 + random()
row = [v1, v2]
data.append(row)
data_exog = [x + random() for x in range(100)]
# fit model
model = VARMAX(data, exog=data_exog, order=(1, 1))
model_fit = model.fit(disp=False)
# make prediction
data_exog2 = [[100]]
yhat = model_fit.forecast(exog=data_exog2)
print(yhat)
```

More Information

- statsmodels.tsa.statespace.varmax.VARMAX API
- statsmodels.tsa.statespace.varmax.VARMAXResults
- Vector autoregression on Wikipedia

Simple Exponential Smoothing (SES)

The Simple Exponential Smoothing (SES) method models the next time step as an exponentially weighted linear function of observations at prior time steps.

The method is suitable for univariate time series without trend and seasonal components.

Python Code

```
# SES example
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = SimpleExpSmoothing(data)
model_fit = model.fit()
# make prediction
```

```python
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

More Information

- statsmodels.tsa.holtwinters.SimpleExpSmoothing API
- statsmodels.tsa.holtwinters.HoltWintersResults API
- Exponential smoothing on Wikipedia

Holt Winter’s Exponential Smoothing (HWES)

The Holt Winter’s Exponential Smoothing (HWES) also called the Triple Exponential Smoothing method models the next time step as an exponentially weighted linear function of observations at prior time steps, taking trends and seasonality into account.

The method is suitable for univariate time series with trend and/or seasonal components.

Python Code

```python
# HWES example
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = ExponentialSmoothing(data)
model_fit = model.fit()
# make prediction
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

More Information

- statsmodels.tsa.holtwinters.ExponentialSmoothing API
- statsmodels.tsa.holtwinters.HoltWintersResults API
- Exponential smoothing on Wikipedia

Further Reading

This section provides more resources on the topic if you are looking to go deeper.

- Statsmodels: Time Series analysis API
- Statsmodels: Time Series Analysis by State Space Methods

Summary

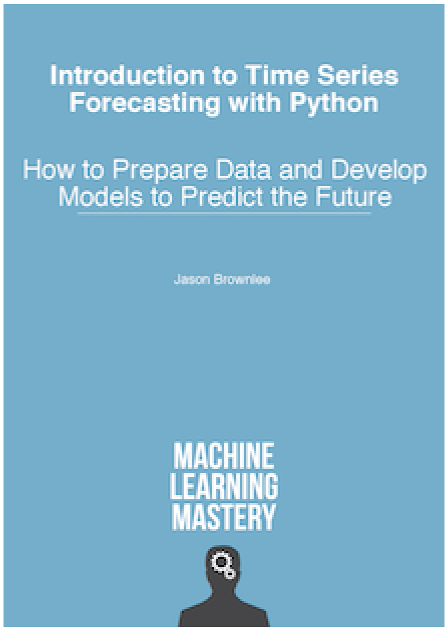
In this post, you discovered a suite of classical time series forecasting methods that you can test and tune on your time series dataset.

Did I miss your favorite classical time series forecasting method?
Let me know in the comments below.

Did you try any of these methods on your dataset?
Let me know about your findings in the comments.

Do you have any questions?
Ask your questions in the comments below and I will do my best to answer.

Want to Develop Time Series Forecasts with Python?


Introduction to Time Series Forecasting with Python
How to Prepare Data and Develop Models to Predict the Future
Jason Brownlee
MACHINE LEARNING MASTERY

Develop Your Own Forecasts in Minutes

...with just a few lines of python code

Discover how in my new Ebook:
Introduction to Time Series Forecasting With Python

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

Finally Bring Time Series Forecasting to Your Own Projects

Skip the Academics. Just Results.

Click to learn more.

Tweet Share Share

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 →

< Statistics for Machine Learning (7-Day Mini-Course)

Taxonomy of Time Series Forecasting Problems >

172 Responses to *11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)*

Adriena Welch August 6, 2018 at 3:20 pm #

REPLY ↩

Hi Jason, thanks for such an excellent and comprehensive post on time series. I sincerely appreciate your effort. As you ask for the further topic, just wondering if I can request you for a specific topic I have been struggling to get an output. It's about Structural Dynamic Factor model (SDFM) by Barigozzi, M., Conti, A., and Luciani, M. (Do euro area countries respond asymmetrically to the common monetary policy) and Mario Forni Luca Gambetti (The Dynamic Effects of Monetary Policy: A Structural Factor Model Approach). Would it be possible for you to go over and estimate these two models using Python or R? It's just a request from me and sorry if it doesn't go with your interest.

Jason Brownlee August 7, 2018 at 6:23 am #

REPLY ↩

Thanks for the suggestion. I've not heard of that method before.

Kamal Singh August 6, 2018 at 6:19 pm #

REPLY ↩

I am working on Time series or Prediction with neural network and SVR, I want to this in matlab by scratch can you give me the references of this materials
Thank you in advance

Jason Brownlee August 7, 2018 at 6:26 am #

REPLY ↩

Sorry, I don't have any materials for matlab, it is only really used in universities.

Catalin August 6, 2018 at 8:50 pm #

REPLY ↩

Hi Jason! From which editor do you import the python code into the webpage of your article? Or what kind of container it that windowed control used to display the python code?

Jason Brownlee August 7, 2018 at 6:26 am #

REPLY ↩

Great question, I explain the software I use for the website here:
https://machinelearningmastery.com/faq/single-faq/what-software-do-you-use-to-run-your-website

Mike August 7, 2018 at 2:28 am # REPLY

Thanks for all the things to try!

I recently stumbled over some tasks where the classic algorithms like linear regression or decision trees outperformed even sophisticated NNs. Especially when boosted or averaged out with each other.

Maybe its time to try the same with time series forecasting as I’m not getting good results for some tasks with an LSTM.

Jason Brownlee August 7, 2018 at 6:30 am # REPLY

Always start with simple methods before trying more advanced methods.

The complexity of advanced methods just be justified by additional predictive skill.

Elie Kawerk August 7, 2018 at 2:36 am # REPLY

Hi Jason,

Thanks for this nice post!

You’ve imported the sin function from math many times but have not used it.

I’d like to see more posts about GARCH, ARCH and co-integration models.

Best,
Elie

Jason Brownlee August 7, 2018 at 6:30 am # REPLY

Thanks, fixed.

I have a post on ARCH (and friends) scheduled.

Elie Kawerk August 7, 2018 at 2:38 am # REPLY

Will you consider writing a follow-up book on advanced time-series models soon?

Jason Brownlee August 7, 2018 at 6:32 am #

REPLY ↩

Yes, it is written. I am editing it now. The title will be “Deep Learning for Time Series Forecasting”.

CNNs are amazing at time series, and CNNs + LSTMs together are really great.

Elie Kawerk August 7, 2018 at 6:40 am #

REPLY ↩

will the new book cover classical time-series models like VAR, GARCH, ..?

Jason Brownlee August 7, 2018 at 2:29 pm #

REPLY ↩

The focus is deep learning (MLP, CNN and LSTM) with tutorials on how to get the most from classical methods (Naive, SARIMA, ETS) before jumping into deep learning methods. I hope to have it done by the end of the month.

Elie Kawerk August 7, 2018 at 5:02 pm #

This is great news! Don’t you think that R is better suited than Python for classical time-series models?

Jason Brownlee August 8, 2018 at 6:15 am #

Perhaps generally, but not if you are building a system for operational use. I think Python is a better fit.

Dark7wind August 9, 2018 at 7:16 am #

Great to hear this news. May I ask if the book also cover the topic of multivariate and multistep?

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

Yes, there are many chapters on multi-step and most chapters work with multivariate data.

Ger July 6, 2019 at 5:36 pm #

Well, although it is a really helpful and useful book as it is usually made by Jason, this book does not cover multivariate time series problems, in fact Jason explicitely says "This is not a treatment of complex time series problems. It does
not provide tutorials on advanced topics like multi-step sequence forecasts, multivariate time series problems or spatial-temporal prediction problems."

Jason Brownlee July 7, 2019 at 7:49 am #

Correct. I cover complex topics in "deep learning for time series forecasting".

Søren August 7, 2018 at 10:27 pm #

REPLY ↩

Sounds amazing that you finally 😉 are geting the new book out on time-series models – when will it be available to buy?

Jason Brownlee August 8, 2018 at 6:20 am #

REPLY ↩

Thanks. I hope by the end of the month or soon after.

Kalpesh Ghadigaonkar March 5, 2019 at 1:01 am #

REPLY ↩

Hi, Can you help me with Arimax ?

Arun Mishra August 10, 2018 at 5:25 am #

REPLY ↩

I use Prophet.
https://facebook.github.io/prophet/docs/quick_start.html

Also, sometimes FastFourier Transformations gives a good result.

Jason Brownlee August 10, 2018 at 6:21 am #

REPLY ↩

Thanks.

AJ Rader August 16, 2018 at 7:11 am #

REPLY ↩

I would second the use of prophet, especially in the context of shock events — this is where this approach has a unique advantage.

Jason Brownlee August 16, 2018 at 1:56 pm #

REPLY ↩

Thanks for the suggestion.

Naresh May 4, 2019 at 4:28 pm #

REPLY ↩

Hi,can you pls help to get the method for timeseries forecasting of10000 products at same time .

Jason Brownlee May 5, 2019 at 6:24 am #

REPLY ↩

I have some suggestions here that might help (replace “site” with “product”):
https://machinelearningmastery.com/faq/single-faq/how-to-develop-forecast-models-for-multiple-sites

User104 May 17, 2019 at 2:24 pm #

REPLY ↩

Hi Arun,
Can you let me know how you worked with fbprophet. I am struggling with the installation of fbprophet module. Since it’s asking for c++ complier. Can you please share how you installed the c++ complier. I tried all ways to resolve it.

Thanks

Jason Brownlee May 17, 2019 at 2:53 pm #

REPLY ↩

I have not worked with fbprophet, sorry.

TJ Slezak June 5, 2019 at 1:45 am #

REPLY

conda install gcc

Jason Brownlee June 5, 2019 at 8:47 am #

Nice!

Ravi Rokhade August 10, 2018 at 5:19 pm #

REPLY

What are the typical application domain of these algos?

Jason Brownlee August 11, 2018 at 6:06 am #

REPLY

Forecasting a time series across domains.

Alberto Garcia Galindo August 11, 2018 at 12:14 am #

REPLY

Hi Jason!
Firstly I congratulate you for your blog. It is helping me a lot in my final work on my bachelor's degree in Statistics!
What are the assumptions for make forecasting on time series using Machine Learning algorithms? For example, it must to be stationary? Thanks!

Jason Brownlee August 11, 2018 at 6:11 am #

REPLY

Gaussian error, but they work anyway if you violate assumptions.

The methods like SARIMA/ETS try to make the series stationary as part of modeling (e.g. differencing).

You may want to look at power transforms to make data more Gaussian.

Neeraj August 12, 2018 at 4:55 pm #

REPLY

Hi Jason
I'm interested in forecasting the temperatures
I'm provided with the previous data of the temperature
Can you suggest me the procedure I should follow in order to solve this problem

Jason Brownlee August 13, 2018 at 6:15 am #

REPLY ↩

Yes, an SARIMA model would be a great place to start.

Den August 16, 2018 at 12:15 am #

REPLY ↩

Hey Jason,

Cool stuff as always. Kudos to you for making me a ML genius!

Real quick:
How would you combine VARMAX with an SVR in python?

Elaboration.
Right now I am trying to predict a y-value, and have x1…xn variables.
The tricky part is, the rows are grouped.
So, for example.

If the goal is to predict the price of a certain car in the 8th year, and I have data for 1200 cars, and for each car I have x11_xnm –> y1_xm data (meaning that let's say car_X has data until m=10 years and car_X2 has data until m=3 years, for example).

First I divide the data with the 80/20 split, trainset/testset, here the first challenge arises. How to make the split?? I chose to split the data based on the car name, then for each car I gathered the data for year 1 to m. (If this approach is wrong, please tell me) The motivation behind this, is that the 80/20 could otherwise end up with data of all the cars of which some would have all the years and others would have none of the years. aka a very skewed distribution.

Then I create a model using an SVR, with some parameters.
And then I try to predict the y-values of a certain car. (value in year m)

However, I do not feel as if I am using the time in my prediction. Therefore, I turned to VARMAX.

Final question(s).
How do you make a time series prediction if you have multiple groups [in this case 1200 cars, each of which have a variable number of years(rows)] to make the model from?
Am I doing right by using the VARMAX or could you tell me a better approach?

Sorry for the long question and thank you for your patience!

Best,

Den

Jason Brownlee August 16, 2018 at 6:09 am #

REPLY

You can try model per group or across groups. Try both and see what works best.

Compare a suite of ml methods to varmax and use what performs the best on your dataset.

Petrônio Cândido August 16, 2018 at 6:36 am #

REPLY

Hi Jason!

Excellent post! I also would like to invite you to know the Fuzzy Time Series, which are data driven, scalable and interpretable methods to analyze and forecast time series data. I have recently published a python library for that on http://petroniocandido.github.io/pyFTS/ .

All feedbacks are welcome! Thanks in advance!

Jason Brownlee August 16, 2018 at 1:55 pm #

REPLY

Thanks for sharing.

SuFian AhMad May 15, 2019 at 3:43 pm #

REPLY

Hello Sir, Can you please share an example code using your Fuzzy Logic timeseries library..
I want to implement Fuzzy Logic time series, and i am just a student, so that's why it will be a great help from you if you will help me in this.
I just need a sample code that is written in python.

Chris Phillips August 30, 2018 at 8:19 am #

REPLY

Hi Jason,

Thank you so much for the many code examples on your site. I am wondering if you can help an amatur like me on something.

When I pull data from our database, I generally do it for multiple SKU's at the same time into a large table. Considering that there are thousands of unique SKU's in the table, is there a methodology you would recommend for generating a forecast for each individual SKU? My initial thought is to run a loop and say something to the effect of: For each in SKU run…Then the VAR Code or the SARIMA code.

Ideally I'd love to use SARIMA, as I think this works the best for the data I am looking to forecast, but if that is only available to one SKU at a time and VAR is not constrained by this, it will work as well. If there is a better methodology that you know of for these, I would gladly take this advice as well!

Thank you so much!

Jason Brownlee August 30, 2018 at 4:49 pm #

REPLY ↩

Yes, I'd encourage you to use this methodology:
https://machinelearningmastery.com/how-to-develop-a-skilful-time-series-forecasting-model/

Eric September 6, 2018 at 6:32 am #

REPLY ↩

Great post. I'm currently investigating a state space approach to forecasting. Dynamic Linear Modeling using a Kálmán Filter algorithm (West, Hamilton). There is a python package, pyDLM, that looks promising, but it would be great to hear your thoughts on this package and this approach.

Jason Brownlee September 6, 2018 at 2:07 pm #

REPLY ↩

Sounds good, I hope to cover state space methods in the future. To be honest, I've had limited success but also limited exposure with the methods.

Not familiar with the lib. Let me know how you go with it.

Alex Rodriguez April 18, 2019 at 7:30 am #

REPLY ↩

Indeed it's an excellent lib.

I use it almost everyday and it really improved the effectiveness of my forecasts over any other method.

Roberto Tomás September 27, 2018 at 7:38 am #

REPLY ↩

Hi Jason, I noticed using VARMAX that I had to remove seasonality — enforcing stationarity .. now I have test and predictions data that I cannot plot (I can, but it doesn't look right _at all_). I'm wondering if there are any built-ins that handle translation to and from seasonality for me? My notebook is online:
https://nbviewer.jupyter.org/github/robbiemu/location-metric-data/blob/master/appData%20and%20locationData.ipynb

Jason Brownlee September 27, 2018 at 2:43 pm #

REPLY ↩

Typically I would write a function to perform the transform and a sister function to invert it.

I have examples here:
https://machinelearningmastery.com/machine-learning-data-transforms-for-time-series-forecasting/

Does that help?

Sara October 2, 2018 at 7:36 am #

REPLY

Thanks for your great tutorial posts. This one was very helpful. I am wondering if there is any method that is suitable for multivariate time series with a trend or/and seasonal components?

Jason Brownlee October 2, 2018 at 11:03 am #

REPLY

Yes, you can try MLPs, CNNs and LSTMs.

You can experiment with each with and without data prep to make the series stationary.

Sara October 3, 2018 at 1:48 am #

REPLY

Thanks for your respond. I also have another question I would appreciate if you help me. I have a dataset which includes multiple time series variables which are not stationary and seems that these variables are not dependent on each other. I tried ARIMA for each variable column, also VAR for the pair of variables, I expected to get better result with ARIMA model (for non-stationarity of time series) but VAR provides much better prediction. Do you have any thought why?

Jason Brownlee October 3, 2018 at 6:20 am #

REPLY

No, go with the method that gives the best performance.

Eric October 17, 2018 at 9:52 am #

REPLY

Hi Jason,

In the (S/V)ARIMAX procedure, should I check to see if my exogenous regressors are stationary and difference if them if necessary before fitting?

Y = data2 = [x + random() for x in range(101, 200)]
X = data1 = [x + random() for x in range(1, 100)]

If I don’t, then I can’t tell if a change in X is related to a change in Y, or if they are both just trending with time. The time trend dominates as 0 <= random() <= 1

In R, Hyndman recommends "[differencing] all variables first as estimation of a model with non-stationary errors is not consistent and can lead to “spurious regression”".

https://robjhyndman.com/hyndsight/arimax/

Does SARIMAX handle this automatically or flag me if I have non-stationary regressors?

Thanks

Jason Brownlee October 17, 2018 at 2:27 pm #

REPLY ↩

No, the library will not do this for you. Differencing is only performed on the provided series, not the exogenous variables.

Perhaps try with and without and use the approach that results in the lowest forecast error for your specific dataset.

Andrew K October 23, 2018 at 9:09 am #

REPLY ↩

Hi Jason,

Thank you for this wonderful tutorial.

I do have a question regarding data that isn’t continuous, for example, data that can only be measured during daylight hours. How would you approach a time series analysis (forecasting) with data that has this behavior? Fill non-daylight hour data with 0’s or nan’s?

Thanks.

Jason Brownlee October 23, 2018 at 2:25 pm #

REPLY ↩

I’d encourage you to test many different framings of the problem to see what works.

If you want to make data contiguous, I have some ideas here:
https://machinelearningmastery.com/handle-missing-timesteps-sequence-prediction-problems-python/

Khalifa Ali October 23, 2018 at 4:48 pm #

REPLY ↩

Hey..
Kindly Help us in making hybrid forecasting techniques.
Using two forecasting technique and make a hybrid technique from them.

Like you may use any two techniques mentioned above and make a hybrid technique form them.
Thanks.

Jason Brownlee October 24, 2018 at 6:25 am # REPLY

Sure, what problem are you having with using multiple methods exactly?

Mohammad Alzyout October 31, 2018 at 6:25 pm # REPLY

Thank you for your excellent and clear tutorial.

I wondered which is the best way to forecast the next second Packet Error Rate in DSRC network for safety messages exchange between vehicles to decide the best distribution over Access Categories of EDCA.

I hesitated to choose between LSTM or ARMA methodology.

Could you please guide me to the better method of them ?

Kindly, note that I'm beginner in both methods and want to decide the best one to go deep with it because I don't have enouph time to learn both methods especially they are as I think from different backgrounds.

Thank you in advance.

Best regards,
Mohammad.

Jason Brownlee November 1, 2018 at 6:03 am # REPLY

I recommend testing a suite of methods in order to discover what works best for your specific problem.

Jawad November 8, 2018 at 12:33 am # REPLY

Hi Jason,
Thanks for great post. I have 2 questions. First, is there a way to calculate confidence intervals in HWES, because i could not find any way in the documentation. And second, do we have something like 'nnetar' R's neural network package for time series forecasting available in python.
Regards

Jason Brownlee November 8, 2018 at 6:10 am # REPLY

I'm not sure if the library has a built in confidence interval, you could calculate it yourself:
https://machinelearningmastery.com/confidence-intervals-for-machine-learning/

What is "nnetar"?

Jawad Iqbal November 22, 2018 at 8:46 am #

REPLY

Thanks for your reply Jason. "nnetar" is a function in R,
https://www.rdocumentation.org/packages/forecast/versions/8.4/topics/nnetar
it is used for time series forecasting. I could not find anything similar in Python.
but now i am using your tutorial of LSTM for time series forecasting.
And i am facing an issue that my data points are 750. and when i do prediction the way you have mentioned i.e. feed the one step forecast back to the new forecast step. So, the plot of my forecasting is just the repetition of my data. Forecast look just like the cyclic repetition of the training data. I don't know what am i missing.

Jason Brownlee November 22, 2018 at 2:08 pm #

REPLY

Perhaps try this tutorial:
https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

Rima December 4, 2018 at 9:59 pm #

REPLY

Hi Jason,
Thank you for this great post!
In VARMAX section, at the end you wrote:
"The method is suitable for univariate time series without trend and seasonal components and exogenous variables."
I understand from the description of VARMAX that it takes as input, multivariate time series and exogenous variables. No?
Another question, can we use the seasonal_decompose (https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal.seasonal_decompose.html) function in python to remove the seasonality and transform our time series to stationary time series? If so, is the result residual (output of seasonal_decompose) is what are we looking for?

Thanks!
Rima

Jason Brownlee December 5, 2018 at 6:16 am #

REPLY

Thanks, fixed.

Rima December 11, 2018 at 9:36 pm #

REPLY ↩

What about Seasonal_decompose method? Do we use residual result or the trend?

Jason Brownlee December 12, 2018 at 5:53 am #

REPLY ↩

Sorry, I don’t understand, perhaps you can elaborate your question?

Rima December 12, 2018 at 7:36 pm #

The seasonal_decompose function implemented in python gives us 4 resutls: the original data, the seasonal component, the trend component and the residual component. Which component should we use to forecast this curve? the residual or the trend component?

Jason Brownlee December 13, 2018 at 7:50 am #

I generally don’t recommend using the decomposed elements in forecasting. I recommend performing the transforms on your data yourself.

Lucky December 5, 2018 at 12:49 am #

REPLY ↩

Hi Jason,

Could you please help me list down the names of all the models available to forecast a univariate time series?

Thanks!

Jason Brownlee December 5, 2018 at 6:18 am #

REPLY ↩

Does the above post not help?

Jane December 6, 2018 at 5:21 am #

REPLY

Hi Jason,

Thank you this was super helpful!

For the AR code, is there any modification I can make so that model predicts multiple periods as opposed to the next one? For example, if am using a monthly time series, and have data up until August 2018, the AR predicts September 2018. Can it predict September 2018, October, 2018, and November 2018 based on the same model and give me these results?

Jason Brownlee December 6, 2018 at 6:03 am #

REPLY

Yes, you can specify the interval for which you need a prediction.

Jane December 7, 2018 at 3:29 am #

REPLY

How might I go about doing that? I have read through the statsmodel methods and have not found a variable that allows this

Jason Brownlee December 7, 2018 at 5:25 am #

REPLY

The interval is specified either to the forecast() or the predict() method, I given an example here that applies to most statsmodels forecasting methods:
https://machinelearningmastery.com/make-sample-forecasts-arima-python/

Esteban December 21, 2018 at 6:56 am #

REPLY

Hi, thank you so much for your post.
I have a question, have you used or have you any guidelines for the use of neural networks in forescating time series, using CNN and LSTMboth together?

Jason Brownlee December 21, 2018 at 3:15 pm #

REPLY

Yes, I have many examples and a book on the topic. You can get started here:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

mk December 22, 2018 at 10:34 pm # REPLY

All methods have common problems. In real life, we do not need to predict the sample data. The sample data already contains the values of the next moment. The so-called prediction is only based on a difference, time lag. That is to say, the best prediction is performance delay. If we want to predict the future, we don't know the value of the current moment. How do we predict? Or maybe we have collected the present and past values, trained for a long time, and actually the next moment has passed. What need do we have to predict?

Jason Brownlee December 23, 2018 at 6:06 am # REPLY

You can frame the problem any way you wish, e.g. carefully define what inputs you have and what output you want to predict, then fit a model to achieve that.

Dr. Omar January 2, 2019 at 1:40 am # REPLY

Dear Jason : your post and book look interesting , I am interested in forecasting a daily close price for a stock market or any other symbol, data collected is very huge and contain each price (let's say one price for each second) , can you briefly tell how we can predict this in general and if your book and example codes if applied will yield to future data.
can we after inputting our data and producing the plot for the past data , can we extend the time series and get the predicted priced for next day/month /year , please explain

Jason Brownlee January 2, 2019 at 6:41 am # REPLY

This is a common question that I answer here:
https://machinelearningmastery.com/faq/single-faq/can-you-help-me-with-machine-learning-for-finance-or-the-stock-market

AD January 4, 2019 at 7:51 pm # REPLY

Hi Jason,

Thank you for this great post.
I have a requirement of predicting receipt values for open invoices of various customers. I am taking closed invoices – whose receipt amount is used to create training data and open invoices as test data.

Below is the list of columns I will be getting as raw data
For Test Data – RECEIPT_AMOUNT, RECEIPT_DATE will be blank, depicting Open Invoices
For Training Data – Closed Invoices will have receipt amount and receipt date

CUSTOMER_NUMBER
CUSTOMER_TRX_ID
INVOICE_NUMBER
INVOICE_DATE
RECEIPT_AMOUNT
BAL_AMOUNT
CUSTOMER_PROFILE
CITY_STATE
STATE
PAYMENT_TERM
DUE_DATE
PAYMENT_METHOD
RECEIPT_DATE

It would be a great help if you can guide me which algo be suitable for this requirement. I think a multivariate method can satisfy this requirement

Thanks,
AD

Jason Brownlee January 5, 2019 at 6:53 am # REPLY ↩

I recommend the following process for new predictive modeling problems:
https://machinelearningmastery.com/start-here/#process

Marius January 8, 2019 at 7:17 am # REPLY ↩

Hi Jason,

Are STAR models relevant here as well?

Kindest
Marius

Jason Brownlee January 8, 2019 at 11:12 am # REPLY ↩

What are star models?

Kostyantyn Kravchenko June 27, 2019 at 8:53 pm # REPLY ↩

Hi Jason, STAR models are Space-Time Autoregression models. I have the same question. I have a multivariate time-series with additional spatial dimension: latitude and longitude. So we need to account not only for the time lags but also for the spacial interactions. I'm trying to find a clear example in Python with no luck so far…

Jason Brownlee June 28, 2019 at 6:01 am #

REPLY ↩

What are you're inputs and outputs exactly?

Heracles January 11, 2019 at 8:35 pm #

REPLY ↩

Hi Jason,

Thanks for this.
I want to forecast whether an event would happen or not. Would that SARMAR actually work work if we have a binary column in it?
How would I accomplish something like this including the time?

Jason Brownlee January 12, 2019 at 5:40 am #

REPLY ↩

Sounds like it might be easier to model the problem as time series classification.

I have some examples of activity recognition, which is time series classification that might provide a good starting point:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

Gary Morton January 13, 2019 at 5:00 am #

REPLY ↩

Good morning

A quality cheat sheet for time series, which I took time to re-create and decided to try an augment by adding code snippets for ARCH and GARH

It did not take long to realize that Statsmodels does not have an ARCH function, leading to a google search that took me directly to:

https://machinelearningmastery.com/develop-arch-and-garch-models-for-time-series-forecasting-in-python/

Great work =) Thought to include here as I did not see a direct link, sans your above comment on thinking to do an ARCH and GARCH module.

also for reference:

LSTM time series model
https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/

MLP and Keras Time Series
https://machinelearningmastery.com/time-series-prediction-with-deep-learning-in-python-with-keras/

Cheers and thank you
-GM

Jason Brownlee January 13, 2019 at 5:45 am #

REPLY ↩

Thanks, and many more here, but neural nets are not really “classic” methods and arch only forecasts volatility:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

Mahmut COLAK January 14, 2019 at 10:08 am #

REPLY ↩

Hi Jason,

Thank you very much this paper. I have a time series problem but i can’t find any technique for applying. My dataset include multiple input and one output like multiple linear regression but also it has timestamp. Which algorithm is the best solution for my problem?

Thanks.

Jason Brownlee January 14, 2019 at 11:15 am #

REPLY ↩

I have many exmaples you can get started here:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

Mahmut COLAK January 14, 2019 at 10:16 am #

REPLY ↩

Hi Jason,

I have a problem about time series data.
My dataset include multiple input and one output.
Normally it is like multiple linear regression but as additional has timestamp 🙁
So i can’t find any solution or algorithm.
For example: AR, MA, ARIMA, ARIMAX, VAR, SARIMAX or etc.
Which one is the best for my problem?

Thanks.

Jason Brownlee January 14, 2019 at 11:15 am # REPLY ↩

I recommend testing a suite of methods and discover what works best for your specific dataset.

RedzCh January 18, 2019 at 11:12 pm # REPLY ↩

one thing is there any methods to do grouped forecasting by keys or category so you have lots of forecasts , there is this on R to an extent

Jason Brownlee January 19, 2019 at 5:41 am # REPLY ↩

I'm not sure I follow, can you elaborate please?

Rodrigo January 18, 2019 at 11:15 pm # REPLY ↩

First of all, I have read two of your books(Basics_for_Linear_Algebra_for_Machine_Learning and deep_learning_time_series_forecasting) and the simplicity with which you explain difficult concepts is brilliant. I'm using the second one to face the problem hat I present below.
I'm facing a predicting problem for food alerts. The goal is to predict the variables of the most probable alert in the next x days (also any information I could get about future alerts is really useful for me).Alerts are recorded over time (so it's a time series problem).
The problem is that observations are not uniform over time (not separated by equal time lapses), i.e: since alerts are only recorded when they happen, there can be one day without alerts and another with 50 alerts. As you indicate in your book, it is a discontiguous time series.

The entry for the possible model could be the alerts (each alert correctly coded as they are categorical variables) of the last x days, but this entry must have a fixed size/format. Since the time windows don't have the same number of alerts, I don't know what is the correct way to deal with this problem.

Any data formatting suggestion to make the observations uniform over time?

Or should I just face the problem in a different way (different inputs)?

Thank you for your great work.

Jason Brownlee January 19, 2019 at 5:44 am # REPLY ↩

Sounds like a great problem!

There are many ways to frame and model the problem and I would encourage you to explore a number and discover what works best.

First, you need to confirm that you have data that can be used to predict the outcome, e.g. is it temporally dependent, or whatever it is dependent upon, can the model get access to that.

Then, perhaps explore modeling it as a time series classification problem, e.g. is the even going to occur in this interval. Explore different interval sizes and different input history sizes and see what works.

Let me know how you go.

Mery January 28, 2019 at 2:33 am # REPLY

Hello Sir,

Thank you for these information

I have a question.

I wanna know if we can use the linear regression model for time series data ?

Jason Brownlee January 28, 2019 at 7:15 am # REPLY

You can, but it probably won't perform as well as specalized linear methods like SARIMA or ETS.

sunil February 5, 2019 at 5:40 pm # REPLY

I have time series data,i want to plot the seasonality graph from it. I am familiar with holt-winter. Are there any other methods?

Jason Brownlee February 6, 2019 at 7:38 am # REPLY

You can plot the series directly to see any seasonality that may be present.

Sai Teja February 11, 2019 at 9:11 pm # REPLY

Hi Jason Brownlee
like auto_arima function present in the R ,Do we have any functions like that in python
for VAR,VARMAX,SES,HWES etc

Jason Brownlee February 12, 2019 at 8:00 am #

REPLY ↩

Yes, I have written a few examples. Perhaps start here:
https://machinelearningmastery.com/how-to-grid-search-sarima-model-hyperparameters-for-time-series-forecasting-in-python/

Michal February 13, 2019 at 1:08 am #

REPLY ↩

Thank you, this list is great primer!

Jason Brownlee February 13, 2019 at 8:00 am #

REPLY ↩

I’m glad it was helpful.

Bilal February 15, 2019 at 8:41 am #

REPLY ↩

Dear Jason,

Thanks for your valuable afford and explanations in such a simple way…

What about the very beginning models of
– Cumulative
– Naive
– Holt’s
– Dampened Holt’s
– Double ESM

I would be very good to see the structural developments of the code from simple to more complex one.

Thank you very much in advance.

Best regards,

Bilal

Jason Brownlee February 15, 2019 at 2:17 pm #

REPLY ↩

Thanks for the suggestion.

Ayoub Benaissa March 2, 2019 at 8:44 am #

REPLY ↰

I was told to build a bayesian regression forecast
which one of these is the best?
because I did not understand “bayesian regression” meaning

Jason Brownlee March 2, 2019 at 9:37 am #

REPLY ↰

Perhaps ask the person who gave you the assignment what they meant exactly?

Rafael March 12, 2019 at 12:46 am #

REPLY ↰

Thank you, i have a datset en csv format and i can open it in excel, it has data since 1984 until 2019, I want to train an artificial neural network in python i order to make frecastig or predictions about that dataset in csv format, I was thinking in a MLP, coul you help me Jason, a guide pls. Many thanks.

Jason Brownlee March 12, 2019 at 6:54 am #

REPLY ↰

Sounds great, you can get started here:
https://machinelearningmastery.com/start-here/#deep_learning_time_series

Venkat March 24, 2019 at 5:00 pm #

REPLY ↰

Dear Jason,

Thank you for your well written quick great info.

Working on one of the banking use cases i.e. Current account and Saving account attrition prediction.
We are using the last 6 months data for training, we need to predict customers whose balance will reduce more than 70% with one exception, as long money invested in the same bank it is fine.
Great, if you could suggest, which models or time series models will be the best options to try in this case?

Jason Brownlee March 25, 2019 at 6:42 am #

REPLY ↰

I recommend this process:
https://machinelearningmastery.com/how-to-develop-a-skilful-time-series-forecasting-model/

Augusto O. Rodero March 29, 2019 at 8:08 pm # REPLY

Hi Jason,

Thank you so much for this post I learned a lot. I am a fan of the ARMAX models in my work as a hydrologist for streamflow forecasting.

I hope you can share something about Gamma autoregressive models or GARMA models which work well even for non-Gaussian time series which the streamflow time series mostly are. Can we do GARMA in python?

Jason Brownlee March 30, 2019 at 6:26 am # REPLY

Thanks for the suggestion, I'll look into the method.

Zack March 30, 2019 at 11:18 pm # REPLY

This blog is very helpful to a novice like me. I have been running the examples you have provided with some changes to create seasonality for example (period of 10 as in 0 to 9, back to 0, again to 9 with randomness thrown in). Linear regression seems to be better at it than the others which I find surprising. What am I missing?

Jason Brownlee March 31, 2019 at 9:30 am # REPLY

You're probably not missing anything.

If a simpler model works, use it!

Andrew April 1, 2019 at 10:01 pm # REPLY

Hi Jason,
Thank you for all your posts, they are so helpful for people who are starting in this area. I am trying to forecast some data and they recommended me to use NARX, but I haven't found a good implementation in python. Do you know other method implemented in python similar to NARX?

Jason Brownlee April 2, 2019 at 8:11 am # REPLY

You can use a SARIMAX as a NARX, just turn off all the aspects you don't need.

Berta April 19, 2019 at 10:19 pm #

REPLY ↩

Hi Jason,

Thank you for all you share us. it's very helpful.

Jason Brownlee April 20, 2019 at 7:38 am #

REPLY ↩

I'm happy to hear that.

jessy April 20, 2019 at 11:30 am #

REPLY ↩

sir,
above 11 models are time series forecasting models, in few section you are discussing about persistence models…what is the difference.

Jason Brownlee April 21, 2019 at 8:18 am #

REPLY ↩

Persistence is a naive model, e.g. "no model".

Naomi May 4, 2019 at 1:55 am #

REPLY ↩

Very good work. Thanks for sharing.

The line in VARMAX "The method is suitable for multivariate time series without trend and seasonal components and exogenous variables." is very confusing.

I guess you mean no trend no seaonal but with exogenous?

Jason Brownlee May 4, 2019 at 7:12 am #

REPLY ↩

Yes, fixed. Thanks!

User104 May 17, 2019 at 2:20 pm #

REPLY ↩

Hi Jason,
Thanks for the post. It was great and easy to understand for a beginner in time series.

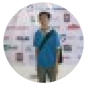
I have data of past 4 years of number of users logged in for a day .
I want to predict the number of users for a day per month.
I used ARIMA but I am getting RMSE 2749 and R2 score 60% .

Can you please suggest methods to increase the accuracy as well as RMSE.

Thanks

Jason Brownlee May 17, 2019 at 2:53 pm # REPLY ↩

Perhaps try some alternate configurations for the ARIMA?
Perhaps try using SARIMA to capture any seasonality?
Perhaps try ETS?



Menghok June 4, 2019 at 9:40 pm # REPLY ↩

Hello Jason, thanks for your explanation.

I have a question. What if my data is time series with multiple variables including categorical data, which model should be used for this? For example, i'm predicting The Air pollution level using the previous observation value of Temperature + Outlook (rain or not).

Thank you.

Jason Brownlee June 5, 2019 at 8:41 am # REPLY ↩

It is a good idea to encode categorical data, e.g. with an integer, one hot encoding or embedding.

Ramesh July 31, 2019 at 2:34 pm # REPLY ↩

Hi Menghok, did you get any luck in implementing forecasting problem when you have one more categorical variable in dataset

Prasanna June 17, 2019 at 3:45 pm # REPLY ↩

Hi Jason,
Thanks for the great post again, wonderful learning experience.
Do you have R codes for the time series methods you described in your article?

or Can you suggest me a good source where can i get R codes to learn some of these methods?

Thanks

Jason Brownlee June 18, 2019 at 6:32 am #

REPLY

Sorry, I don't have R code for time series, perhaps you can start here:
https://machinelearningmastery.com/books-on-time-series-forecasting-with-r/

Ibrahim Rashid June 18, 2019 at 8:39 pm #

REPLY

Thanks for the post. Please do check out AnticiPy which is an open-source tool for forecasting using Python and developed by Sky.

The goal of AnticiPy is to provide reliable forecasts for a variety of time series data, while requiring minimal user effort.

AnticiPy can handle trend as well as multiple seasonality components, such as weekly or yearly seasonality. There is built-in support for holiday calendars, and a framework for users to define their own event calendars. The tool is tolerant to data with gaps and null values, and there is an option to detect outliers and exclude them from the analysis.

Ease of use has been one of our design priorities. A user with no statistical background can generate a working forecast with a single line of code, using the default settings. The tool automatically selects the best fit from a list of candidate models, and detects seasonality components from the data. Advanced users can tune this list of models or even add custom model components, for scenarios that require it. There are also tools to automatically generate interactive plots of the forecasts (again, with a single line of code), which can be run on a Jupyter notebook, or exported as .html or .png files.

Check it out here:
https://pypi.org/project/anticipy/

Jason Brownlee June 19, 2019 at 7:52 am #

REPLY

Thanks for the note.

Min June 19, 2019 at 7:46 am #

REPLY

Hi Jason,

Thanks for this great post!

I have a question for time series forecasting. Have you heard about Dynamic Time Warping? As far as I know, this is a method for time series classification/clustering, but I think it can also be used for forecasting based on the similar time series. What do you think about this method compared to ARIMA? Do you think it will be better if I combine both two methods? For example, use DTW to group similar time series and then use ARIMA for each group?

Thanks

Jason Brownlee June 19, 2019 at 8:20 am #

REPLY

I don't have any posts on the topic, but I hope to cover it in the future.

Ida4526 June 22, 2019 at 7:11 am #

REPLY

Can you please explain why you use len(data) in your predict arguments? I was using the .forecast feature for a while which is for out of sample forecasts but I keep getting an error on my triple expo smoothing. Apparently, the .predict can be used for in-sample prediction as well as out of sample. The arguments are start and end, and you use len(data) for both which is confusing me. Will this really forecast or will it just produce a forecast for months in the past?

Jason Brownlee June 22, 2019 at 7:43 am #

REPLY

Great question.

To predict the next or index beyond the known data.

More details here:
https://machinelearningmastery.com/make-sample-forecasts-arima-python/

Ida4526 June 25, 2019 at 1:03 am #

REPLY

Thanks for the reply! I was reading through the explanation in your linked article and it was great. Can the .predict() do multiple periods in the future like .forecast? I was using .forecast(12) for forecasting 12 months into the future.

Ida4526 June 25, 2019 at 1:17 am #

REPLY

EDIT: Dumb question- did not read until the end. If you got time, check out my stack post though: https://stackoverflow.com/questions/56709745/statsmodels-operands-could-not-be-broadcast-together-in-pandas-series . I think i found some error in statsmodels as the error is thrown in the .forecast function as it wants me to specify a frequency. I found the potential misstep by reading through the actual code for ExponentialSmoothing and it was the only part that really referenced the frequency. Its either that, or my noob is really showing.

Jason Brownlee June 25, 2019 at 6:27 am #

Perhaps this code example will help:
https://machinelearningmastery.com/how-to-grid-search-triple-exponential-smoothing-for-time-series-forecasting-in-python/

karim June 23, 2019 at 9:05 pm #

REPLY ↩

Hi Jason,

Thanks for all of your awesome tutorial.

Can you please provide any link of your tutorial which has described forecasting of multivariate time series with a statistical model like VAR?

Jason Brownlee June 24, 2019 at 6:27 am #

REPLY ↩

I do not have a tutorial on VAR, sorry.

karim June 25, 2019 at 7:00 am #

REPLY ↩

OK, thanks for your reply. Hopefully, if you can manage time will come with a tutorial on VAR for us. 🙂

qwizak July 2, 2019 at 10:56 pm #

REPLY ↩

Hi Jason, do you cover all these models using a real dataset in your book?

Jason Brownlee July 3, 2019 at 8:33 am #

REPLY ↩

I focus on AR/ARIMA in the intro book and SARIMA/ETS+deep learning in the deep learning time series book.

Shabnam July 3, 2019 at 11:26 pm #

REPLY ↩

Hi Jason,

Thanks for the helpful tutorial. I've been trying to solve a sales forecast problem but haven't been any successful. The data is the monthly records of product purchases (counts) with their respective prices for ten years. The records do not show either a significant auto-correlation for a wide range of lags or seasonality. The records are stationary though.

Among the time series models, I have tried (S)ARIMA, exponential methods, the Prophet model, and a simple LSTM. I have also tried regression models using a number of industrial and financial indices and the product price. Unfortunately, no method has led to an acceptable result. With regression models, the test R^2 is always negative.

My questions are:
* What category of problems is this problem more relevant to?
* Do you have any suggestions for possibly suitable approaches to follow for this kind of problems?

Thank you in advance.

Shabnam

Jason Brownlee July 4, 2019 at 7:49 am #

REPLY ↩

Perhaps the time series is not predictable?

Shabnam July 8, 2019 at 11:04 pm #

REPLY ↩

I guess that might be the case. I'm also guessing that maybe I don't have sufficiently relevant explanatory variables to obtain a good regression model. Thanks for your feedback though. And, thanks again for your very helpful tutorials.

Shabnam

Jason Brownlee July 9, 2019 at 8:11 am #

REPLY ↩

You're welcome.

George July 9, 2019 at 1:05 am #

REPLY ↩

Hello Jason,
Do you know if there any way to randomly sample a fitted VARMA time series using the statsmodel library.
Just like using the sm.tsa.arma_generate_sample for the ARMA.
I cant seem to see this how this is done anywhere.

Jason Brownlee July 9, 2019 at 8:12 am #

REPLY ↩

Not off hand, sorry.

Manjinder Singh July 11, 2019 at 9:06 pm #

REPLY ↩

Hello Jason,
It is really nice and informative article.
I am having network data and in that data there are different parameters (network incidents) which slows down the network. From the lot of parameters one is network traffic. I have DATE and TIME when network traffic crosses a certain threshold at certain location. I need to draw a predictive model so that i can spot when network traffic is crossing threshold value at a particular location, so that i can take preventive measures prior to occurrence of that parameter at that place.
So please can you suggest me appropriate model for this problem.

Jason Brownlee July 12, 2019 at 8:41 am #

REPLY ↩

Perhaps it might be interesting to explore modeling the problem as a time series classification?

Juan Flores July 11, 2019 at 9:56 pm #

REPLY ↩

Hi, Jason,

We've been having trouble with statsmodels' ARIMA. It just doesn't work and takes forever. What can you tell us about these issues? Do you know of any alternatives to statsmodels?

ThX,

Juan

Jason Brownlee July 12, 2019 at 8:43 am #

REPLY ↩

Perhaps try a sklearn linear regression model directly?

Juan J. Flores July 13, 2019 at 10:47 pm #

REPLY

Dear Jason,

Thanks for the answer.

Of course there are many regression models available in sklearn. The point is that statsmodels seems to fail miserably, both in time and accuracy. That is, with respect to their arima (family) set of functions.

Question is if you know an alternative python library providing that missing functionality. R works well. Mathematica even better. At the moment, my students are working on interfacing with R, since we have not found a sound Python library for arima.

Have a good one.

Juan

Jason Brownlee July 14, 2019 at 8:11 am #

REPLY

I see, good question.

I have found the statsmodels implementation to be reliable, but only if the data is sensible and only if the order is modest.

I cannot recommend another library at this time. R may be slightly more reliable, but is not bulletproof.

Manjinder Singh July 12, 2019 at 7:12 pm #

REPLY

In time series classification when I am plotting it is showing day wise data . For an instance if i am taking data of past 1 month and applying Autoregression time series classification on it then i am not able to fetch detailed outputs from that chart. From detailed output I mean that 1 incident is occuring number of times in a day so I want visibility in such a way so that everytime an incident occur it should be noticed on that plot.
Is there any precise way by which I can do it.I would be really thankful if you would help me.
Thank you

Jason Brownlee July 13, 2019 at 6:54 am #

REPLY

Perhaps resample the data to daily before or after modeling, depending on whether you want model data this way or only view forecasts this way.

Berns B. August 8, 2019 at 9:40 am #

REPLY

I can’t seem to make VAR work? Gives me a lot of errors?

```
ValueError Traceback (most recent call last)
in ()
11 # fit model
12 model = VAR(data)
—> 13 model_fit = model.fit()
14 # make prediction
15 yhat = model_fit.forecast(model_fit.y, steps=1)

D:\Users\Berns\Anaconda3\envs\time_series_p27\lib\site-packages\statsmodels\tsa\vector_ar\var_model.pyc in fit(self, maxlags, method, ic, trend, verbose)
644 self.data.xnames[k_trend:])
645
–> 646 return self._estimate_var(lags, trend=trend)
647
648 def _estimate_var(self, lags, offset=0, trend=’c’):

D:\Users\Berns\Anaconda3\envs\time_series_p27\lib\site-packages\statsmodels\tsa\vector_ar\var_model.pyc in _estimate_var(self, lags, offset, trend)
666 exog = None if self.exog is None else self.exog[offset:]
667 z = util.get_var_endog(endog, lags, trend=trend,
–> 668 has_constant=’raise’)
669 if exog is not None:
670 # TODO: currently only deterministic terms supported (exoglags==0)

D:\Users\Berns\Anaconda3\envs\time_series_p27\lib\site-packages\statsmodels\tsa\vector_ar\util.pyc in
get_var_endog(y, lags, trend, has_constant)
36 if trend != ‘nc’:
37 Z = tsa.add_trend(Z, prepend=True, trend=trend,
—> 38 has_constant=has_constant)
39
40 return Z

D:\Users\Berns\Anaconda3\envs\time_series_p27\lib\site-packages\statsmodels\tsa\tsatools.pyc in
add_trend(x, trend, prepend, has_constant)
97 col_const = x.apply(safe_is_const, 0)
98 else:
—> 99 ptp0 = np.ptp(np.asanyarray(x), axis=0)
100 col_is_const = ptp0 == 0
101 nz_const = col_is_const & (x[0] != 0)
```

```
D:\Users\Berns\Anaconda3\envs\time_series_p27\lib\site-packages\numpy\core\fromnumeric.pyc in ptp(a, axis, out, keepdims)
2388 else:
2389 return ptp(axis=axis, out=out, **kwargs)
-> 2390 return _methods._ptp(a, axis=axis, out=out, **kwargs)
2391
2392

D:\Users\Berns\Anaconda3\envs\time_series_p27\lib\site-packages\numpy\core\_methods.pyc in _ptp(a, axis, out, keepdims)
151 def _ptp(a, axis=None, out=None, keepdims=False):
152 return um.subtract(
–> 153 umr_maximum(a, axis, None, out, keepdims),
154 umr_minimum(a, axis, None, None, keepdims),
155 out

ValueError: zero-size array to reduction operation maximum which has no identity
```

Jason Brownlee August 8, 2019 at 2:21 pm #

REPLY ↩

Sorry, I'm not sure about the cause of your error.

Perhaps confirm statsmodels is up to date?

Berns B. August 8, 2019 at 12:25 pm #

REPLY ↩

By the way Doc Jason, I just bought your book today this morning. I have a huge use case for a time series at work I can use them for.

Jason Brownlee August 8, 2019 at 2:21 pm #

REPLY ↩

Thanks, you have a lot of fun ahead!

I'm here to help if you have questions, email me directly:
https://machinelearningmastery.com/contact/

Berns B. August 8, 2019 at 1:19 pm #

REPLY ↩

Darn indentions! Now code works.

```
# VAR example
from statsmodels.tsa.vector_ar.var_model import VAR
from random import random
# contrived dataset with dependency
data = list()
for i in range(100):
v1 = i + random()
v2 = v1 + random()
row = [v1, v2]
data.append(row)
# fit model
model = VAR(data)
model_fit = model.fit()
# make prediction
yhat = model_fit.forecast(model_fit.y, steps=1)
print(yhat)
```

Jason Brownlee August 8, 2019 at 2:22 pm #

REPLY ↰

I'm very happy to hear that.

More on how copy code from posts here:
https://machinelearningmastery.com/faq/single-faq/how-do-i-copy-code-from-a-tutorial

Leave a Reply

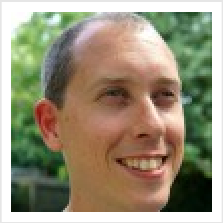
Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome to Machine Learning Mastery!



Hi, I'm **Jason Brownlee**, PhD.
I write tutorials to help developers (*like you*) get results with machine learning.

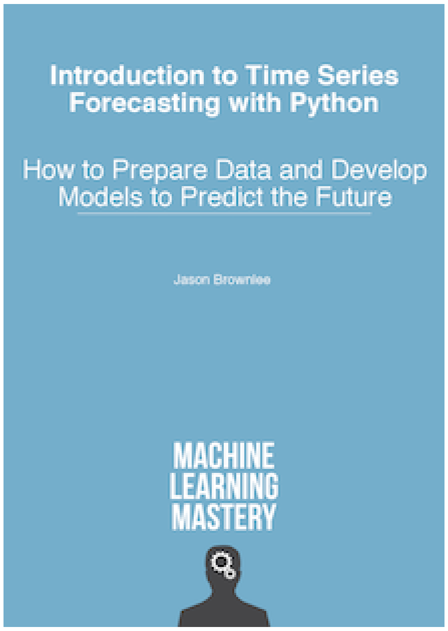
Read More

More Tutorials?

Start Here
(over 780 tutorials!)

Time Series Forecasting with Python
Explore time series data and predict the future.

Click to Get Started Now!



Introduction to Time Series Forecasting with Python

How to Prepare Data and Develop Models to Predict the Future

Jason Brownlee

MACHINE LEARNING MASTERY

Join Newsletter

Subscribe Now
(join over 150,000 readers!)

Address: PO Box 206, Vermont Victoria 3133, Australia. | ACN: 626 223 336.

RSS | Twitter | Facebook | LinkedIn

Privacy | Disclaimer | Terms | Contact | Sitemap | Search