

f (<https://www.facebook.com/AnalyticsVidhya>) **t** (<https://twitter.com/analyticsvidhya>)

g+ (<https://plus.google.com/+Analyticsvidhya/posts>)

in (<https://in.linkedin.com/company/analytics-vidhya>)

u **SUJITDEOKAR30** (<https://id.analyticsvidhya.com/accounts/profile/>)

HOME (<https://www.analyticsvidhya.com>)

BLOG ARCHIVE (<https://www.analyticsvidhya.com/blog-archive/>)

TRAININGS (https://courses.analyticsvidhya.com?utm_source=home_blog_navbar)

DISCUSS (<https://discuss.analyticsvidhya.com>) **DATAHACK** (<https://datahack.analyticsvidhya.com>)

JOBS (<https://www.analyticsvidhya.com/jobs/>)

CORPORATE (<https://www.analyticsvidhya.com/corporate/>)



(<https://www.analyticsvidhya.com/blog/>)



(https://analyticsvidhya.com/datahack-summit-2019/?utm_source=blog&utm_medium=topBanner&utm_campaign=DHS2019)

LEARN ▾ **ENGAGE** ▾ **COMPETE** ▾



AI & ML BLACKBELT ([https://courses.analyticsvidhya.com/bundles/ai-](https://courses.analyticsvidhya.com/bundles/ai-blackbelt-beginner-to-master?utm_source=home_blog_navbar)

BLACKBELT-BEGINNER-TO-MASTER?UTM_SOURCE=HOME BLOG_NAVBAR)

COURSES (<https://courses.analyticsvidhya.com>) ▾

DATAHACK SUMMIT 2019 ([https://www.analyticsvidhya.com/datahack-](https://www.analyticsvidhya.com/datahack-summit-2019?utm_source=home_blog_navbar)

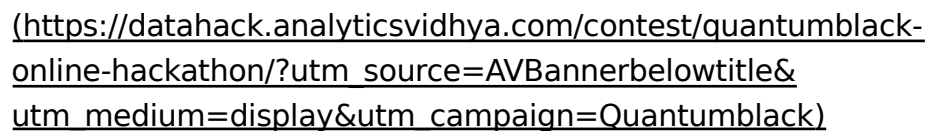
SUMMIT-2019?UTM_SOURCE=HOME BLOG_NAVBAR)

CONTACT ([HTTPS://WWW.ANALYTICSVIDHYA.COM/CONTACT/](https://www.analyticsvidhya.com/contact/))

[MACHINE LEARNING \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/MACHINE-LEARNING/\)](https://www.analyticsvidhya.com/blog/category/machine-learning/)

[PYTHON \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/PYTHON-2/\)](https://www.analyticsvidhya.com/blog/category/python-2/)

[TIME SERIES \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/TIME-SERIES/\)](https://www.analyticsvidhya.com/blog/category/time-series/)

AISHWARYA SINGH ([HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/AUTHOR/A...](https://www.analyticsvidhya.com/blog/author/a...))

The time series data most of us are exposed to deals primarily

Rank	Name
------	------

1	SRK (https://datahack. /user/profile/SRK)
2	Sahil Verma (https://datahack. /user/profile/sahilv)
3	Tezdhar (https://datahack. /user/profile/Tezdh)
4	mark12 (https://datahack. /user/profile/mark)
5	Denis Vorotyntse (https://datahack. /user/profile/teartl)

(https://datahack.analyticsvidhya.com/top-competitor/?utm_source=blog-avoid&utm_medium=web)

with generating forecasts. Whether that's predicting the demand or sales of a product, the count of passengers in an airline or the closing price of a particular stock, we are used to leveraging tried and tested time series techniques for forecasting requirements.



But as the amount of data being generated increases exponentially, so does the opportunity to experiment with new ideas and algorithms. Working with complex time series datasets is still a niche field, and it's always helpful to expand your repertoire to include new ideas.

And that is what I aim to do in the article by introducing you to the novel concept of time series classification. We will first understand what this topic means and its applications in the industry. But we won't stop at the theory part – we'll get our hands dirty by working on a time series dataset and performing binary time series classification. Learning by doing – this will help you understand the concept in a practical manner as well.

If you have not worked on a time series problem before, I highly recommend first starting with some basic forecasting. You can go through the below article for starters:

- [A comprehensive beginner's guide to create a Time Series Forecast \(with Codes in Python\)](#)

POPULAR POSTS

6 Useful Programming Languages for Data Science You Should Learn (that are not R and Python)
(<https://www.analyticsvidhya.com/blog/2019/06/6-useful-programming-languages-data->

[\(https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/\)](https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/)

Table of contents

1. Introduction to Time Series Classification
 1. ECG Signals
 2. Image Data
 3. Sensors
2. Setting up the Problem Statement
3. Reading and Understanding the Data
4. Preprocessing
5. Building our Time Series Classification Model

Introduction to Time Series Classification

Time series classification has actually been around for a while. But it has so far mostly been limited to research labs, rather than industry applications. But there is a lot of research going on, new datasets being created and a number of new algorithms being proposed. When I first came across this time series classification concept, my initial thought was – how can we classify a time series and what does a time series classification data look like? I'm sure you must be wondering the same thing.

As you can imagine, time series classification data differs from a regular classification problem since the attributes have an ordered sequence. Let's have a look at some time series classification use cases to understand this difference.

1) Classifying ECG/EEG signals

ECG, or electrocardiogram, records the electrical activity of the heart and is widely used to diagnose various heart problems. These ECG signals are captured using external

[science-r-python/](#)

24 Ultimate Data Science Projects To Boost Your Knowledge and Skills (& can be accessed freely)

[\(https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/\)](https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/)

Commonly used Machine Learning Algorithms (with Python and R Codes) [\(https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/\)](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/)

7 Regression

Techniques you should know!

[\(https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/\)](https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/)

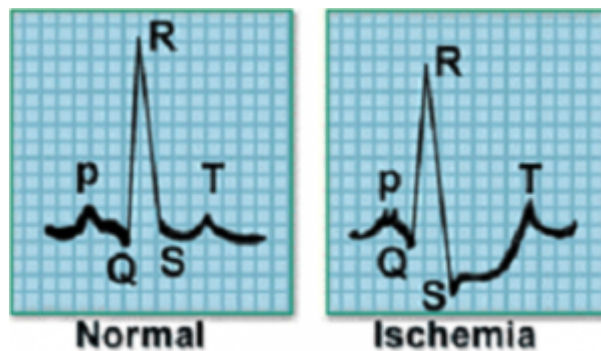
A Complete Python Tutorial to Learn Data Science from Scratch [\(https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/\)](https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/)

Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes)

[/blog/2018/10/predicting-stock-](https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-)

electrodes.

For example, consider the following signal sample which represents the electrical activity for one heartbeat. The image on the left represents a normal heartbeat while the one adjacent to it represents a Myocardial Infarction.



The data captured from the electrodes will be in time series form, and the signals can be classified into different classes. We can also classify EEG signals which record the electrical activity of the brain.

2) Image Classification

Images can also be in a sequential time-dependent format. Consider the following scenario:

Crops are grown in a particular field depending upon the weather conditions, soil fertility, availability of water and other external factors. A picture of this field is taken daily for 5 years and labeled with the name of the crop planted on the field. Do you see where I'm going with this? The images in the dataset are taken after a fixed time interval and have a defined sequence, which can be an important factor in classifying the images.

3) Classifying Motion Sensor Data

Sensors generate high-frequency data that can identify the movement of objects in their range. By setting up multiple wireless sensors and observing the change in signal strength in

price-machine-learningnd-deep-learning-techniques-python/)

Understanding Support Vector Machine algorithm from examples (along with code) (<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>)

Introduction to k-Nearest Neighbors: A powerful Machine Learning Algorithm(with implementation in Python & R) (<https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>)

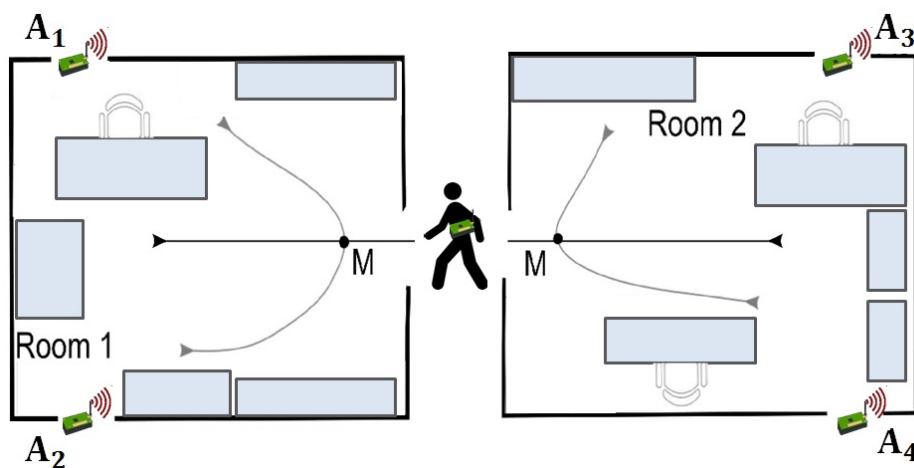
the sensors, we can identify the object's direction of movement.

What other applications can you think of where we can apply time series classification? Let me know in the comments section below the article.

Setting up the Problem Statement

We will be working on the '*Indoor User Movement Prediction*' problem. In this challenge, multiple motion sensors are placed in different rooms and the goal is to identify whether an individual has moved across rooms, based on the frequency data captured from these motion sensors.

There are four motion sensors (A1, A2, A3, A4) placed across two rooms. Have a look at the below image which illustrates where the sensors are positioned in each room. The setup in these two rooms was created in 3 different pairs of rooms (group1, group2, group3).



A person can move along any of the six pre-defined paths shown in the above image. If a person walks on path 2, 3, 4 or 6, he moves within the room. On the other hand, if a person follows path 1 or path 5, we can say that the person has moved between the rooms.

The sensor reading can be used to identify the position of a person at a given point in time. As the person moves in the

RECENT POSTS

10 Artificial Intelligence (AI) Startups in India You Should Know
(<https://www.analyticsvidhya.com/blog/2019/07/10-ai-startups-india/>)

JULY 2, 2019

6 Powerful Open Source Machine Learning GitHub Repositories for Data Scientists
(<https://www.analyticsvidhya.com/blog/2019/07/6-powerful-open-source-machine-learning-github-repositories-data-scientists/>)

JULY 1, 2019

Top 5 Must-Read Answers - What does a Data Scientist do on a Daily Basis?

room or across rooms, the reading in the sensor changes. This change can be used to identify the path of the person.

Now that the problem statement is clear, it's time to get down to coding! In the next section, we will look at the dataset for the problem which should help clear up any lingering questions you might have on this statement. You can download the dataset from this link: **[Indoor User Movement Prediction](https://archive.ics.uci.edu/ml/datasets/Indoor+User+Movement+Prediction+from+RSS+data)** (**<https://archive.ics.uci.edu/ml/datasets/Indoor+User+Movement+Prediction+from+RSS+data>**).

Reading and Understanding the Data

Our dataset comprises of 316 files:

- 314 **MovementAAL** csv files containing the readings from motion sensors placed in the environment
- A **Target** csv file that contains the target variable for each MovementAAL file
- One **Group Data** csv file to identify which MovementAAL file belongs to which setup group
- The **Path** csv file that contains the path which the object took

Let's have a look at the datasets. We'll start by importing the necessary libraries.

```
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
from os import listdir
```

(**<https://www.analyticsvidhya.com/blog/2019/06/what-does-data-scientist-do-daily-basis-top-5-quora-answers/>**)

JUNE 27, 2019

Announcing DataHack Summit 2019 - The Biggest Artificial Intelligence and Machine Learning Conference Yet (**<https://www.analyticsvidhya.com/blog/2019/06/announcing-datahack-summit-2019-biggest-ai-machine-learning-conference/>**)

JUNE 26, 2019


```

from keras.preprocessing import sequence
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM

from keras.optimizers import Adam
from keras.models import load_model
from keras.callbacks import ModelCheckpoint

```

Before loading all the files, let's take a quick sneak peek into the data we are going to deal with. Reading the first two files from the movement data:

```

df1 = pd.read_csv('/MovementAAL/dataset/MovementAAL_RSS_
1.csv')
df2 = pd.read_csv('/MovementAAL/dataset/MovementAAL_RSS_
2.csv')

```

```
df1.head()
```

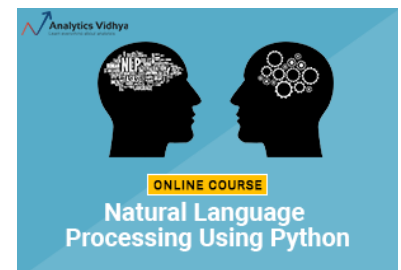
	#RSS_anchor1	RSS_anchor2	RSS_anchor3	RSS_anchor4
0	-0.90476	-0.48	0.28571	0.30
1	-0.57143	-0.32	0.14286	0.30
2	-0.38095	-0.28	-0.14286	0.35
3	-0.28571	-0.20	-0.47619	0.35
4	-0.14286	-0.20	0.14286	-0.20

```
df2.head()
```

	#RSS_anchor1	RSS_anchor2	RSS_anchor3	RSS_anchor4
0	-0.57143	-0.20	0.71429	0.50
1	-0.76190	-0.48	0.76190	-0.25
2	-0.85714	-0.60	0.85714	0.55
3	-0.76190	-0.40	0.71429	0.60



(http://www.edvancer.in/certified-data-scientist-with-python-course?utm_source=AV&utm_medium=AVads&utm_campaign=AVadsnonfc&utm_content=pythonavad)



Learn to Solve
Text Classification Problems Using NLP

(https://courses.analyticsvidhya.com/courses/natural-language-processing-nlp?utm_source=Sticky_banner&utm_medium=display&utm_campaign=NLPcourse)



(https://courses.analyticsvidhya.com/courses/computer-vision-using-deep-learning-version2?utm_source=Sticky_banner&utm_medium=display&utm_campaign=CVcourse)

4	-0.76190	-0.84	0.85714	0.45
---	----------	-------	---------	------

utm_campaign=CVcourse)

```
df1.shape, df2.shape
```

```
((27, 4), (26, 4))
```

The files contain normalized data from the four sensors - A1, A2, A3, A4. The length of the csv files (number of rows) vary, since the data corresponding to each csv is for a different duration. To simplify things, let us suppose the sensor data is collected every second. The first reading was for a duration of 27 seconds (so 27 rows), while another reading was for 26 seconds (so 26 rows).

We will have to deal with this varying length before we build our model. For now, we will read and store the values from the sensors in a list using the following code block:

```
path = 'MovementAAL/dataset/MovementAAL_RSS_'
sequences = list()
for i in range(1,315):
    file_path = path + str(i) + '.csv'
    print(file_path)
    df = pd.read_csv(file_path, header=0)
    values = df.values
    sequences.append(values)
```

```
targets = pd.read_csv('MovementAAL/dataset/MovementAAL_t
arget.csv')
targets = targets.values[:,1]
```

We now have a list 'sequences' that contains the data from the motion sensors and 'targets' which holds the labels for the csv files. When we print `sequences[0]`, we get the values of sensors from the first csv file:

```
sequences[0]
```

```
array([[ -0.90476 , -0.48      ,  0.28571 ,  0.3      ],
       [ -0.57143 , -0.32      ,  0.14286 ,  0.3      ],
       [ -0.38095 , -0.28      , -0.14286 ,  0.35     ],
       [ -0.28571 , -0.2       , -0.47619 ,  0.35     ],
       [ -0.14286 , -0.2       ,  0.14286 , -0.2      ],
       [ -0.14286 , -0.2       ,  0.047619,  0.       ],
       [ -0.14286 , -0.16      , -0.38095 ,  0.2      ],
       [ -0.14286 , -0.04      , -0.61905 , -0.2      ],
       [ -0.095238, -0.08      ,  0.14286 , -0.55     ],
       [ -0.047619,  0.04      , -0.095238,  0.05     ],
       [ -0.19048 , -0.04      ,  0.095238,  0.4       ],
       [ -0.095238, -0.04      , -0.14286 ,  0.35     ],
       [ -0.33333 , -0.08      , -0.28571 , -0.2      ],
       [ -0.2381   ,  0.04      ,  0.14286 ,  0.35     ],
       [  0.       ,  0.08      ,  0.14286 ,  0.05     ],
       [ -0.095238,  0.04      ,  0.095238,  0.1       ],
       [ -0.14286 , -0.2       ,  0.14286 ,  0.5       ],
       [ -0.19048 ,  0.04      , -0.42857 ,  0.3       ],
       [ -0.14286 , -0.08      , -0.2381   ,  0.15     ],
       [ -0.33333 ,  0.16      , -0.14286 , -0.8       ],
       [ -0.42857 ,  0.16      , -0.28571 , -0.1      ],
       [ -0.71429 ,  0.16      , -0.28571 ,  0.2       ],
       [ -0.095238, -0.08      ,  0.095238,  0.35     ],
       [ -0.28571 ,  0.04      ,  0.14286 ,  0.2       ],
       [  0.       ,  0.04      ,  0.14286 ,  0.1       ],
       [  0.       ,  0.04      , -0.047619, -0.05     ],
       [ -0.14286 , -0.6       , -0.28571 , -0.1      ]])
```

As mentioned previously, the dataset was collected in three different pairs of rooms – hence three groups. This information can be used to divide the dataset into train, test and validation sets. We will load the **DatasetGroup** csv file now:

```
groups = pd.read_csv('MovementAAL/groups/MovementAAL_Dat
asetGroup.csv', header=0)
groups = groups.values[:,1]
```

We will take the data from the first two sets for training purposes and the third group for testing.

Preprocessing Steps

Since the time series data is of varying length, we cannot directly build a model on this dataset. So how can we decide the ideal length of a series? There are multiple ways in which we can deal with it and here are a few ideas (I would love to hear your suggestions in the comment section):

- Pad the shorter sequences with zeros to make the length of all the series equal. In this case, we will be feeding incorrect data to the model
- Find the maximum length of the series and pad the sequence with the data in the last row
- Identify the minimum length of the series in the dataset and truncate all the other series to that length. However, this will result in a huge loss of data
- Take the mean of all the lengths, truncate the longer series, and pad the series which are shorter than the mean length

Let's find out the minimum, maximum and mean length:

```
len_sequences = []
for one_seq in sequences:
    len_sequences.append(len(one_seq))
pd.Series(len_sequences).describe()
```

```
count    314.000000
mean      42.028662
std       16.185303
min       19.000000
25%       26.000000
50%       41.000000
75%       56.000000
max      129.000000
dtype: float64
```

Most of the files have lengths between 40 to 60. Just 3 files are coming up with a length more than 100. Thus, taking the minimum or maximum length does not make much sense. The 90th quartile comes out to be 60, which is taken as the length

of sequence for the data. Let's code it out:

```
#Padding the sequence with the values in last row to max
length
to_pad = 129
new_seq = []
for one_seq in sequences:
    len_one_seq = len(one_seq)
    last_val = one_seq[-1]
    n = to_pad - len_one_seq

    to_concat = np.repeat(one_seq[-1], n).reshape(4, n).
transpose()
    new_one_seq = np.concatenate([one_seq, to_concat])
    new_seq.append(new_one_seq)
final_seq = np.stack(new_seq)

#truncate the sequence to length 60
from keras.preprocessing import sequence
seq_len = 60
final_seq=sequence.pad_sequences(final_seq, maxlen=seq_l
en, padding='post', dtype='float', truncating='post')
```

Now that the dataset is prepared, we will separate it based on the groups. Preparing the train, validation and test sets:

```
train = [final_seq[i] for i in range(len(groups)) if (groups[i]==2)]
validation = [final_seq[i] for i in range(len(groups)) if groups[i]==1]
test = [final_seq[i] for i in range(len(groups)) if groups[i]==3]

train_target = [targets[i] for i in range(len(groups)) if (groups[i]==2)]
validation_target = [targets[i] for i in range(len(groups)) if groups[i]==1]
test_target = [targets[i] for i in range(len(groups)) if groups[i]==3]
```

```
train = np.array(train)
validation = np.array(validation)
test = np.array(test)

train_target = np.array(train_target)
train_target = (train_target+1)/2

validation_target = np.array(validation_target)
validation_target = (validation_target+1)/2

test_target = np.array(test_target)
test_target = (test_target+1)/2
```

Building a Time Series Classification model

We have prepared the data to be used for an LSTM (Long Short Term Memory) model. We dealt with the variable length sequence and created the train, validation and test sets. Let's build a single layer LSTM network.

Note: You can get acquainted with LSTMs in this [wonderfully](#)

[explained tutorial \(https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/\)](https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/). I would advice you to go through that first as it'll help you understand how the below code works.

```
model = Sequential()  
model.add(LSTM(256, input_shape=(seq_len, 4)))  
model.add(Dense(1, activation='sigmoid'))
```

```
model.summary()
```

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 256)	267264
dense_1 (Dense)	(None, 1)	257
Total params: 267,521		
Trainable params: 267,521		
Non-trainable params: 0		

We will now train the model and monitor the validation accuracy:

```
adam = Adam(lr=0.001)  
chk = ModelCheckpoint('best_model.pkl', monitor='val_acc',  
    save_best_only=True, mode='max', verbose=1)  
model.compile(loss='binary_crossentropy', optimizer=adam,  
    metrics=['accuracy'])  
model.fit(train, train_target, epochs=200, batch_size=128,  
    callbacks=[chk], validation_data=(validation, validation_target))
```

```
#loading the model and checking accuracy on the test data
a
model = load_model('best_model.pkl')

from sklearn.metrics import accuracy_score
test_preds = model.predict_classes(test)
accuracy_score(test_target, test_preds)
```

I got an accuracy score of 0.78846153846153844. It's quite a promising start but we can definitely improve the performance of the LSTM model by playing around with the hyperparameters, changing the learning rate, and/or the number of epochs as well.

End Notes

That brings us to the end of this tutorial. The idea behind penning this down was to introduce you to a whole new world in the time series spectrum in a practical manner.

Personally, I found the preprocessing step the most complex section of all the ones we covered. Yet, it is the most essential one as well (otherwise the whole point of time series data will fail!). Feeding the right data to the model is equally important when working on this kind of challenge.

Here is a really cool time series classification resource which I referred to and found the most helpful:

- Paper on “Predicting User Movements in Heterogeneous Indoor Environments by Reservoir Computing”

I would love to hear your thoughts and suggestions in the comment section below.

You can also read this article on Analytics Vidhya's Android APP




<http://play.google.com/store>


[/apps/details?id=com.analyticsvidhya.android&utm_source=blog_article&utm_campaign=blog&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1\)](https://www.analyticsvidhya.com/apps/details?id=com.analyticsvidhya.android&utm_source=blog_article&utm_campaign=blog&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1)


Share this:

 (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/?share=linkedin&nb=1>)

 (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/?share=facebook&nb=1>)

 (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/?share=twitter&nb=1>)

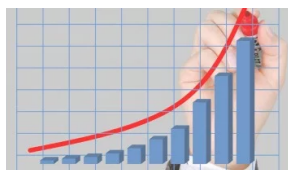
 (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/?share=pocket&nb=1>)

 (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/?share=reddit&nb=1>)

Like this:

Loading...

Related Articles



(<https://www.analyticsvidhya.com/blog/2018/08/auto-arma-time-series-modeling-python-r/>)

[Build High Performance Time Series Models using Auto ARIMA in Python and R](https://www.analyticsvidhya.com/blog/2018/08/auto-arma-time-series-modeling-python-r/)

(<https://www.analyticsvidhya.com/blog/2016/12/top-35-articles-resources-analytics-vidhya-year-2016/>)

[Top 35 Articles and Resources from Analytics Vidhya for the year 2016](https://www.analyticsvidhya.com/blog/2016/12/top-35-articles-resources-analytics-vidhya-year-2016/)

(<https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/>)

[A Gentle Introduction to Handling a Non-Stationary Time Series in Python](https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/)

(<https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/>)

ticsvidhya.com	/blog/2016/12/top-	stationary-time-
/blog/2018/08/auto-	35-articles-	series-python/)
arima-time-series-	resources-analytics-	September 13,
modeling-python-r/)	vidhya-year-2016/)	2018
August 30, 2018	December 29, 2016	In "Python"
In "Python"	In "Machine	
	Learning"	

TAGS : [CLASSIFICATION \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/CLASSIFICATION/\)](https://www.analyticsvidhya.com/blog/tag/classification/), [INDOOR USER MOVEMENT \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/INDOOR-USER-MOVEMENT/\)](https://www.analyticsvidhya.com/blog/tag/indoor-user-movement/), [LSTM \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/LSTM/\)](https://www.analyticsvidhya.com/blog/tag/lstm/), [PYTHON \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/PYTHON/\)](https://www.analyticsvidhya.com/blog/tag/python/), [TIME SERIES \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/TIME-SERIES/\)](https://www.analyticsvidhya.com/blog/tag/time-series/), [TIME SERIES CLASSIFICATION \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/TIME-SERIES-CLASSIFICATION/\)](https://www.analyticsvidhya.com/blog/tag/time-series-classification/)

PREVIOUS ARTICLE

NEXT ARTICLE



**DataHack Radio
#15: Exploring
the Applications
& Potential of
Reinforcement
Learning with
Xander
Steenbrugge**

(<https://www.analyticsvidhya.com/blog/2019/01/datahack-radio-reinforcement-learning-xander/>)



**Build your First
Image
Classification
Model in just 10
Minutes!**

(<https://www.analyticsvidhya.com/blog/2019/01/build-image-classification-model-10-minutes/>)



**Aishwarya Singh****([https://www.analyticsvidhya.com](https://www.analyticsvidhya.com/blog/author/aishwaryasingh/)****[blog/author/aishwaryasingh/](https://www.analyticsvidhya.com/blog/author/aishwaryasingh/))****[/blog/author](https://www.analyticsvidhya.com/blog/author/aishwaryasingh/)****[/aishwaryasingh/](https://www.analyticsvidhya.com/blog/author/aishwaryasingh/))**

An avid reader and blogger who loves exploring the endless world of data science and artificial intelligence. Fascinated by the limitless applications of ML and AI; eager to learn and discover the depths of data science.

10 COMMENTS

**AYAN****[Reply](#)**

January 7, 2019 at 1:38 pm (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-156456>)

I found the tutorial too vague and rushed. Be great if you could clarify on:

- a) What does the target mean in this case?
- b) What does an accuracy 0.78 mean in this scenario?

**AISHWARYA SINGH****[Reply](#)**

January 8, 2019 at 10:44 am
(<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-156469>)

Hi Ayan,

The target variable is to identify whether the person moved from one room to another or did he move within the same room. After training the model on certain data, we used the same to make predictions on unseen data and the accuracy came out to be 78%.

**BHUMIKA****Reply**

February 12, 2019 at 9:05 am

<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-157036>

Hi. How can I predict the time-series patterns that will be present in the data for the upcoming days? Any techniques on that ?

**AISHWARYA SINGH****Reply**

February 20, 2019 at 2:34 pm

<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-157117>

Hi Bhumika,

By pattern do you mean the trend and seasonality of the data? You can use a simple arima model for forecasting, and then decompose to get the trend and seasonality.

**BHUMIKA****Reply**

February 12, 2019 at 9:08 am

<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-157037>

Hi,

How do I classify the patterns in time-series data and set up to predict them in the future based on some input classifiers ?

**MARCEL****Reply**January 7, 2019 at 2:37 pm (<https://www.analyticsvidhya.com>)

</blog/2019/01/introduction-time-series-classification/#comment-156457>)

Thank you Aishwarya,

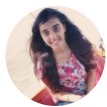
What a nice and thorough introduction.

Well done!

I learned a lot today and it's only 10:00 AM, where I am.

kind regards,

Marcel



AISHWARYA SINGH

[Reply](#)

January 8, 2019 at 10:40 am

<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-156468>)

Really glad you liked it Marcel!



GIANFRANCO

[Reply](#)

January 10, 2019 at 5:27 pm (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-156518>)

Thank you Aishwarya,

Good job !



V V CHAKRADHAR

[Reply](#)

January 14, 2019 at 11:34 am (<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment-156569>)

Any publications on time series classification on Image data ?



AISHWARYA SINGH

[Reply](#)

January 16, 2019 at 2:45 pm

<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/#comment->

156605)

Hi,

Have you checked papers on the website
<http://www.timeseriesclassification.com/>
(<http://www.timeseriesclassification.com/>)

LEAVE A REPLY

Your email address will not be published.

Comment

Name (required)

Email (required)





































Website

SUBMIT COMMENT

☐

Notify me of new posts by email.



ANALYTICS VIDHYA	DATA SCIENTISTS	COMPANIES	JOIN OUR COMMUNITY :
About Us (http://www.analyticsvidhya.com/about-me/)	Blog (https://www.analyticsvidhya.com/blog/)	Post Jobs (https://www.analyticsvidhya.com/jobs/)	 https://www.facebook.com/AnalyticsVidhya/  https://twitter.com/AnalyticsVidhya
Our Team (https://www.analyticsvidhya.com/about-me/team/)	Hackathon (https://datahack.analyticsvidhya.com/)	Trainings (https://trainings.analyticsvidhya.com/)	 https://www.facebook.com/AnalyticsVidhya/  https://twitter.com/AnalyticsVidhya
Career (https://www.analyticsvidhya.com/career-analytics-vidhya/)	Discussions (https://discuss.analyticsvidhya.com/)	Hackathons (https://datahack.analyticsvidhya.com/)	 https://www.facebook.com/AnalyticsVidhya/  https://twitter.com/AnalyticsVidhya
Contact Us (https://www.analyticsvidhya.com/contact/)	Apply Jobs (https://www.analyticsvidhya.com/jobs/)	Advertising (https://www.analyticsvidhya.com/contact/)	 https://www.facebook.com/AnalyticsVidhya/  https://twitter.com/AnalyticsVidhya
Write for us (https://www.analyticsvidhya.com/about-me/write/)	Leaderboard (https://datahack.analyticsvidhya.com/users/)	Reach Us (https://www.analyticsvidhya.com/contact/)	 https://www.facebook.com/AnalyticsVidhya/  https://twitter.com/AnalyticsVidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya
			 https://plus.google.com/+Analyticsvidhya/  https://in.linkedin.com/company/analyticsvidhya

Subscribe to emailer

>

×

-

(http://play.google.com/store/apps/details?id=com.analyticsvidhya.android)

⌵