

[Download Success ROADMAP] To Become a
Data Scientist in 180 days with a JOB
Guarantee*


Register Now (https://bootcamp.analyticsvidhya.com/?utm_source=blog&utm_medium=flashstrip#av-roadmap)


 LOGIN / REGISTER ([HTTPS://ID.ANALYTICSVIDHYA.COM/AUTH/LOGIN/](https://id.analyticsvidhya.com/auth/login/))

NEXT=[HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2020/10/PERFORM-REGRESSION-ANALYSIS-WITH-PYTORCH-SEAMLESSLY/](https://www.analyticsvidhya.com/blog/2020/10/perform-regression-analysis-with-pytorch-seamlessly/))



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

**Certified AI & ML BlackBelt Plus**
50+ Projects | 75+ Mentorship Sessions | 14+ Certifications
[Download Brochure](#)



(https://blackbelt.analyticsvidhya.com/plus?utm_source=blogtopbanner&utm_campaign=Blackbelt)

[INTERMEDIATE \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/INTERMEDIATE/\)](https://www.analyticsvidhya.com/blog/category/intermediate/)

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

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

[STRUCTURED DATA \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/STRUCTURED-DATA/\)](https://www.analyticsvidhya.com/blog/category/structured-data/)

[TECHNIQUE \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/TECHNIQUE/\)](https://www.analyticsvidhya.com/blog/category/technique/)

Perform Regression Analysis with PyTorch Seamlessly!

[GUEST BLOG \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/AUTHOR/GUEST-BLOG/\)](https://www.analyticsvidhya.com/blog/author/guest-blog/), OCTOBER 21, 2020 [LOGIN TO BOOKMARK THI...](#)

Article



Video Book

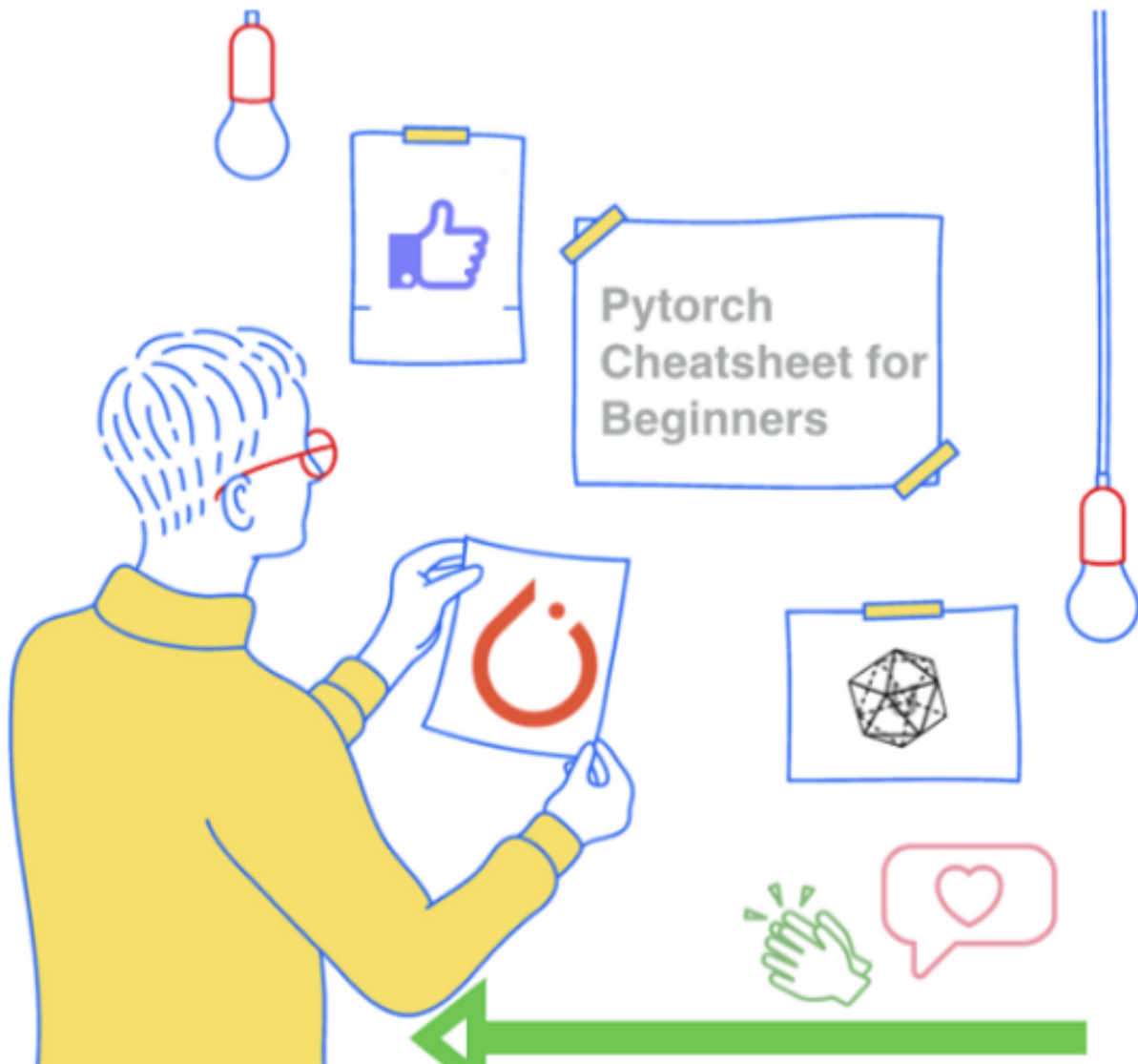


image courtesy.

(<https://www.kdnuggets.com/2019/08/pytorch-cheat-sheet-beginners.html>).

Introduction

This is probably the 1000th article that is going to talk about implementing regression analysis using PyTorch. so how is it different?

Well, before I answer that let me write the series of events that led to this article. So, when I started learning regression in PyTorch, I was excited but I had so many whys and why nots that I got frustrated at one point. So, I thought why not start from scratch- understand the deep learning framework a little better and then delve deep into the complex concepts like CNN, RNN, LSTM, etc. and the easiest way to do so is taking a familiar dataset and explore as much as you can so that you understand the basic building blocks and the key modeling principles.

working principle.

I have tried to explain the modules that are imported, why certain steps are mandatory, and how we evaluate a regression model in PyTorch.

So people, if you have just started or looking for answers as I did, then you are definitely in the right place. 😊

Okay, so let's start with the imports first.

Imports

```
import torch
import torch.nn as nn
```

the torch.nn modules help us create and train neural networks. So definitely we need that. Let's move on:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
```

They look familiar, right? We need them cause we have to do some preprocessing on the dataset we will be using.

Data

The dataset I am going to use is **Celsius to Fahrenheit** data which can be found here: [link](https://www.kaggle.com/domnic/celsius-to-fahrenheit) (<https://www.kaggle.com/domnic/celsius-to-fahrenheit>).

	Celsius	Fahrenheit
0	1169	2136.2
1	1280	2336.0
2	2380	4316.0
3	1508	2746.4
4	2015	3659.0

Data preprocessing step 1: separate out the feature and the label

```
X_train = train_data.iloc[:,0].values
y_train = train_data.iloc[:,1].values
```

Data preprocessing step 2: standardize the data as the values are very large and varied

So if you don't do that for this particular case, then later while training the model, you will likely get inf or nan for loss values, meaning, the model cannot perform backpropagation properly and will result in a faulty model.

```
sc = MinMaxScaler()  
sct = MinMaxScaler()  
X_train=sc.fit_transform(X_train.reshape(-1,1))  
y_train =sct.fit_transform(y_train.reshape(-1,1))
```

we have to make sure that X_train and y_train are 2-d.

Okay, so far so good. Now let's enter into the world of tensors

Data preprocessing Step 3: Convert the numpy arrays to tensors

```
X_train = torch.from_numpy(X_train.astype(np.float32)).view(-1,1)  
y_train = torch.from_numpy(y_train.astype(np.float32)).view(-1,1)
```

The view takes care of the 2d thing in tensor as reshape does in numpy.

Regression Model Building in PyTorch

```
input_size = 1  
output_size = 1
```

input= celsius

output = fahrenheit

Define layer

```
class LinearRegressionModel(torch.nn.Module):

    def __init__(self):
        super(LinearRegressionModel, self).__init__()
        self.linear = torch.nn.Linear(1, 1) # One in and one out

    def forward(self, x):
        y_pred = self.linear(x)
        return y_pred
```

Or we could have simply done this(since it is just a single layer)

```
model = nn.Linear(input_size , output_size)
```

In both cases, we are using `nn.Linear` to create our first linear layer, this basically does a linear transformation on the data, say for a straight line it will be as simple as $y = w \cdot x$, where y is the label and x , the feature. Of course, w is the weight. In our data, celsius and fahrenheit follow a linear relation, so we are happy with one layer but in some cases where the relationship is non-linear, we add additional steps to take care of the non-linearity, say for example add a sigmoid function.

Define loss and optimizer

```
learning_rate = 0.0001

l = nn.MSELoss()

optimizer = torch.optim.SGD(model.parameters(), lr =learning_rate )
```

as you can see, the loss function, in this case, is “**mse**” or “**mean squared error**”. Our goal will be to reduce the loss and that can be done using an optimizer, in this case, **stochastic gradient descent**. That SGD needs initial model parameters or weights and a learning rate.

Okay, now let's start the training.

Training

```
num_epochs = 100
for epoch in range(num_epochs):
    #forward feed
    y_pred = model(X_train.requires_grad_())

    #calculate the loss
    loss= l(y_pred, y_train)

    #backward propagation: calculate gradients
    loss.backward()

    #update the weights
    optimizer.step()

    #clear out the gradients from the last step Loss.backward()
    optimizer.zero_grad()

    print('epoch {}, loss {}'.format(epoch, loss.item()))
```

forward feed: in this phase, we are just calculating the `y_pred` by using some initial weights and the feature values.

loss phase: after the `y_pred`, we need to measure how much prediction error happened. We are using **mse** to measure that.

backpropagation: in this phase gradients are calculated.

step: the weights are now updated.

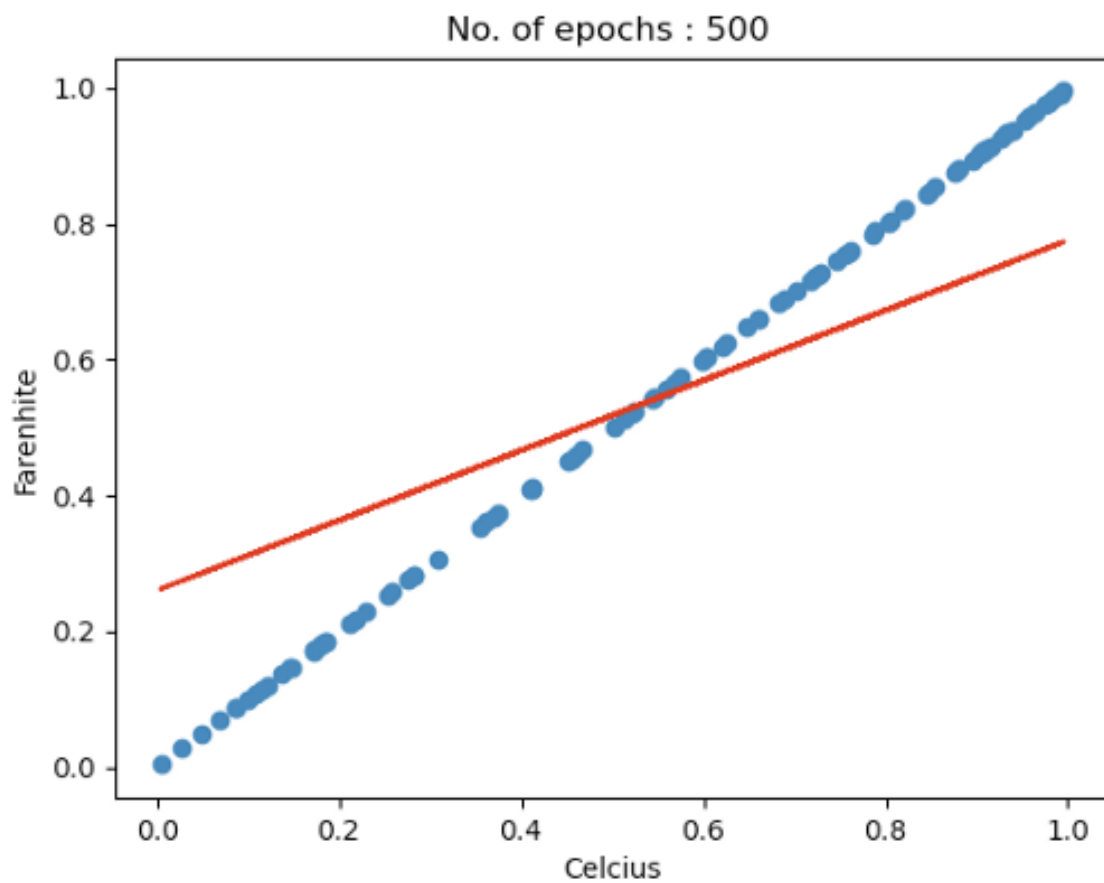
zero_grad: finally, clear the gradients from the last step and make room for the new ones.

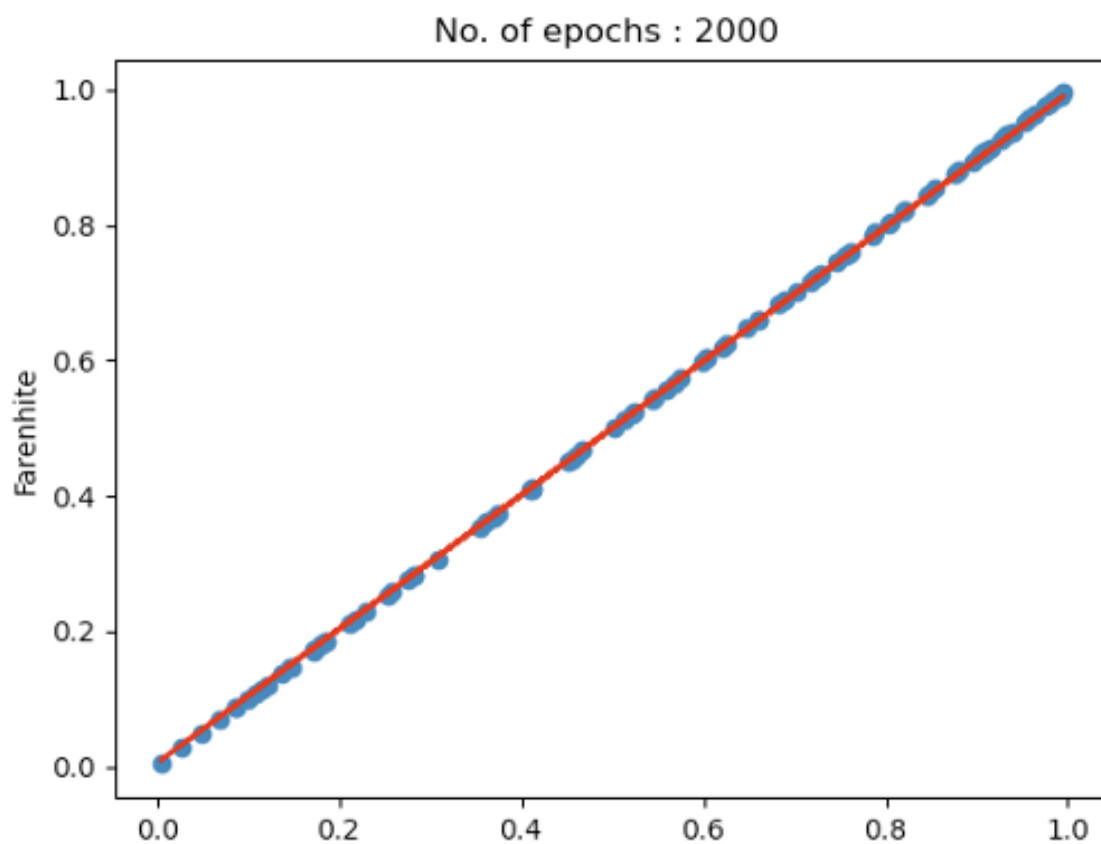
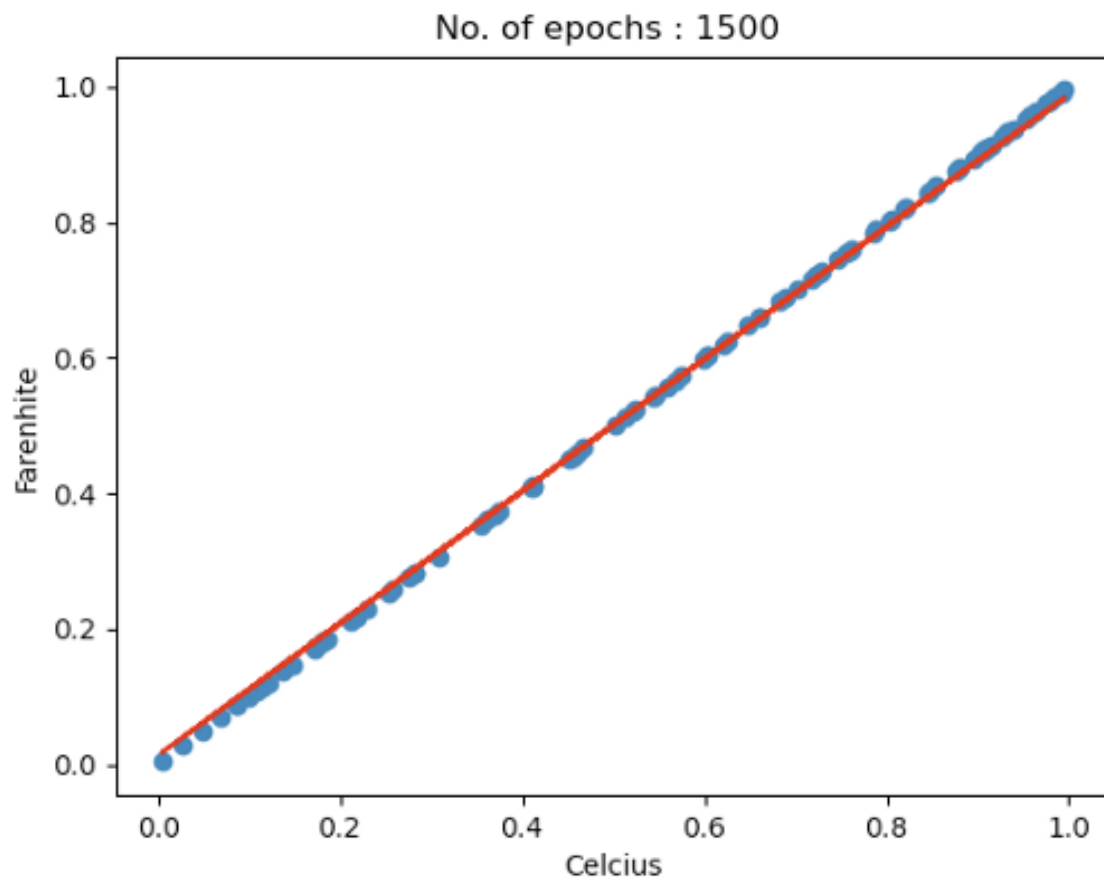
Evaluation

```
predicted = model(X_train).detach().numpy()
```

`detach()` is saying that we do not need to store gradients anymore so detach that from the tensor. Now, let's visualize the model quality with the first 100 data points.

```
plt.scatter(X_train.detach().numpy()[:100] , y_train.detach().numpy()[:100])  
plt.plot(X_train.detach().numpy()[:100] , predicted[:100] , "red")  
plt.xlabel("Celcius")  
plt.ylabel("Farenhite")  
plt.show()
```





Celcius

Notice, how with the increase in the number of the epochs, the predictions are getting better. There are multiple other strategies to optimize the network, for example changing the learning rate, weight initialization techniques, and so on.

Lastly, try with a known celsius value and see if the model is able to predict the fahrenheit value correctly. The values are transformed, so make sure you do a `sc.inverse_transform()` and `sct.inverse_transform()` to get the actual values.

Thank you for reading this article. Please do leave a comment or share your feedback. 😊

References:

https://pytorch.org/tutorials/beginner/nn_tutorial.html
(https://pytorch.org/tutorials/beginner/nn_tutorial.html).

About the Author



I am Dipanwita Mallick, currently working as a Senior Data Scientist at Hewlett Packard Enterprise. My work involves building and productionizing predictive models in the HPC & AI domain to efficiently and effectively use HPC clusters, supercomputers expanded over 10,000 cores with high bandwidth, low latency interconnects, several petabytes of data storage and databases.

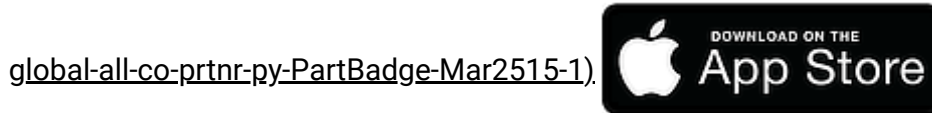
Prior to that I have worked with Cray Inc. in Seattle and United Educators in Washington DC as a Data Scientist.

I have also done my Master's in Data Science from Indiana University, Bloomington.

You can also read this article on our Mobile APP



(https://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://apps.apple.com/us/app/analytics-vidhya/id1470025572>).

Related Articles



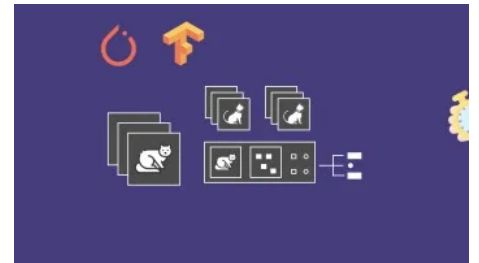
(<https://www.analyticsvidhya.com/blog/2020/12/top-15-free-data-science-courses-to-kick-start-your-data-science-journey/>).

Top 15 Free Data Science Courses to Kick Start your Data Science Journey!
(<https://www.analyticsvidhya.com/blog/2020/12/top-15-free-data-science-courses-to-kick-start-your-data-science-journey/>)



(<https://www.analyticsvidhya.com/blog/2020/07/deploy-an-image-classification-model-using-flask/>).

Deploy an Image Classification Model Using Flask
(<https://www.analyticsvidhya.com/blog/2020/07/deploy-an-image-classification-model-using-flask/>)



(<https://www.analyticsvidhya.com/blog/2020/07/how-to-train-an-image-classification-model-in-pytorch-and-tensorflow/>).

How to Train an Image Classification Model in PyTorch and TensorFlow
(<https://www.analyticsvidhya.com/blog/2020/07/how-to-train-an-image-classification-model-in-pytorch-and-tensorflow/>)



PREPARE FOR CODING INTERVIEW
Placement Preparation Program
www.prepbytes.com



TAGS : [MATPLOTLIB \(https://www.analyticsvidhya.com/blog/tag/matplotlib/\)](https://www.analyticsvidhya.com/blog/tag/matplotlib/), [MIN_MAX SCALER \(https://www.analyticsvidhya.com/blog/tag/min-max-scaler/\)](https://www.analyticsvidhya.com/blog/tag/min-max-scaler/)

<https://www.analyticsvidhya.com/blog/2020/10/perform-regression-analysis-with-pytorch-seamlessly/>

[\(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/MIN_MAX-SCALER/\)](https://www.analyticsvidhya.com/blog/tag/min-max-scaler/), [PANDAS](#)

[\(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/PANDAS/\)](https://www.analyticsvidhya.com/blog/tag/pandas/), [PYTORCH](#)

[\(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/PYTORCH/\)](https://www.analyticsvidhya.com/blog/tag/pytorch/)

NEXT ARTICLE

Introduction and Implementation to Neural Style Transfer – Deep Learning

[\(https://www.analyticsvidhya.com/blog/2020/10/introduction-and-implementation-to-neural-style-transfer-deep-learning/\)](https://www.analyticsvidhya.com/blog/2020/10/introduction-and-implementation-to-neural-style-transfer-deep-learning/)

...

PREVIOUS ARTICLE

What are the differences between Data Lake and Data Warehouse?

[\(https://www.analyticsvidhya.com/blog/2020/10/what-is-the-difference-between-data-lake-and-data-warehouse/\)](https://www.analyticsvidhya.com/blog/2020/10/what-is-the-difference-between-data-lake-and-data-warehouse/)



[\(https://www.analyticsvidhya.com/blog/author/guest-blog/\)](https://www.analyticsvidhya.com/blog/author/guest-blog/)

[Guest Blog \(https://www.analyticsvidhya.com/blog/author/guest-blog/\)](https://www.analyticsvidhya.com/blog/author/guest-blog/)

LEAVE A REPLY

Your email address will not be published.

Comment

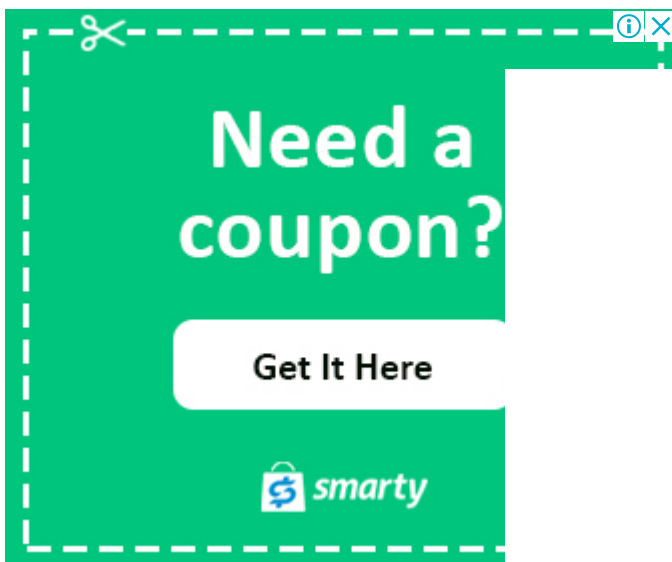
Name (required)

Email (required)

Website

☐ Notify me of new posts by email.

SUBMIT COMMENT



POPULAR POSTS

How to Download Kaggle Datasets using Jupyter Notebook

(<https://www.analyticsvidhya.com/blog/2021/04/how-to-download-kaggle-datasets-using-jupyter-notebook/>)

40 Questions to test a Data Scientist on Clustering Techniques (Skill test Solution)

(<https://www.analyticsvidhya.com/blog/2017/02/test-data-scientist-clustering/>)

Python List Programs For Absolute Beginners (<https://www.analyticsvidhya.com/blog/2021/04/python-list-programs-for-absolute-beginners/>)

Commonly used Machine Learning Algorithms (with Python and R Codes)

(<https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>)

Understanding Delimiters in Pandas read_csv() Function

(https://www.analyticsvidhya.com/blog/2021/04/delimiters-in-pandas-read_csv-function/)

45 Questions to test a data scientist on basics of Deep Learning (along with solution)

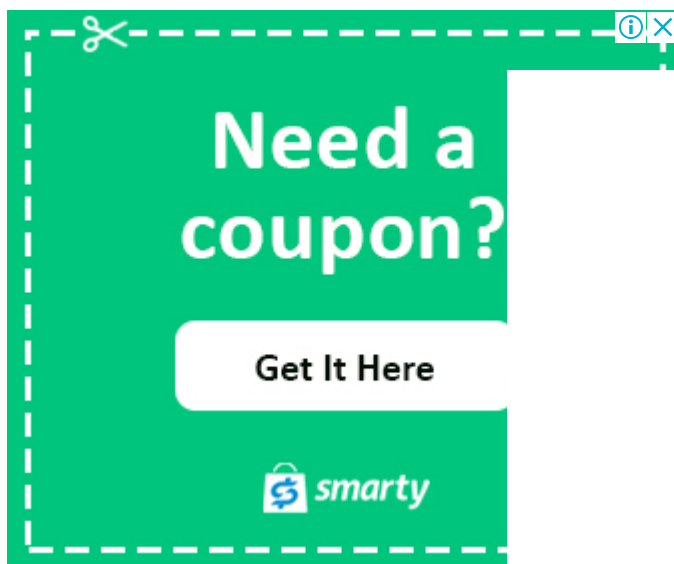
(<https://www.analyticsvidhya.com/blog/2017/01/must-know-questions-deep-learning/>)

30 Questions to test a data scientist on K-Nearest Neighbors (kNN) Algorithm

(<https://www.analyticsvidhya.com/blog/2017/09/30-questions-test-k-nearest-neighbors-algorithm/>)

Introductory guide on Linear Programming for (aspiring) data scientists

(<https://www.analyticsvidhya.com/blog/2017/02/introductory-guide-on-linear-programming-explained-in-simple-english/>)



CAREER RESOURCES

16 Key Questions You Should Answer Before Transitioning into Data Science (<https://www.analyticsvidhya.com/16-key-questions-data->



**science-career-transition/?
&utm_source=Blog&utm_medium=CareerResourceWidget)**

NOVEMBER 23, 2020



**Here's What You Need to Know to Become a Data Scientist!
(<https://www.analyticsvidhya.com/blog/2021/01/heres-what-you-need-to-know-to-become-a-data-scientist/>
&utm_source=Blog&utm_medium=CareerResourceWidget)**

JANUARY 22, 2021



**These 7 Signs Show you have Data Scientist Potential!
(<https://www.analyticsvidhya.com/blog/2020/12/these-7-signs-show-you-have-data-scientist-potential/>
&utm_source=Blog&utm_medium=CareerResourceWidget)**

DECEMBER 3, 2020



**How To Have a Career in Data Science (Business Analytics)?
(<https://www.analyticsvidhya.com/blog/2020/11/how-to-have-a-career-in-data-science-business-analytics/>
&utm_source=Blog&utm_medium=CareerResourceWidget)**

NOVEMBER 26, 2020



**Should I become a data scientist (or a business analyst)?
(<https://www.analyticsvidhya.com/blog/2020/11/become-data-scientist-business-analyst/>
&utm_source=Blog&utm_medium=CareerResourceWidget)**

NOVEMBER 24, 2020

RECENT POSTS

**Spatial Interpolation With and Without Predictor(s)
(<https://www.analyticsvidhya.com/blog/2021/05/spatial-interpolation-with-and-without-predictors/>)**

MAY 21, 2021

**Traditional vs Deep Learning Classification Models | An Empirical Study!
(<https://www.analyticsvidhya.com/blog/2021/05/emperical-study-of-traditional-vs-deep-learning-classification-models/>)**

MAY 21, 2021

Geospatial Analysis | Getting Started With Folium In Python!

(<https://www.analyticsvidhya.com/blog/2021/05/geospatial-analysis-getting-started-with-folium-in-python/>)

MAY 21, 2021

Importance of Cross Validation: Are Evaluation Metrics enough?

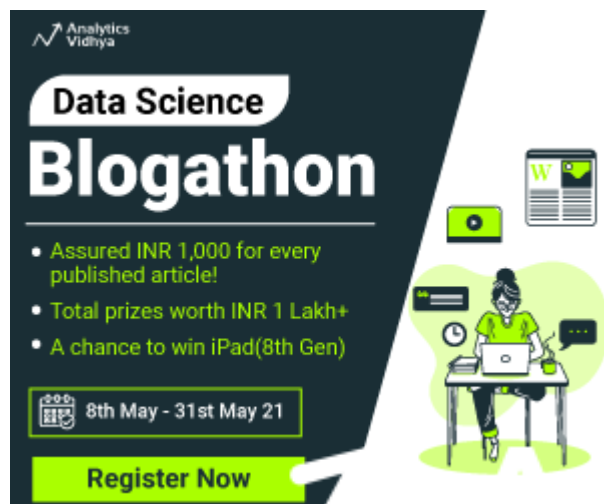
(<https://www.analyticsvidhya.com/blog/2021/05/importance-of-cross-validation-are-evaluation-metrics-enough/>)

MAY 21, 2021



(<https://bootcamp.analyticsvidhya.com/?>

utm_source=Blog&utm_medium=stickybanner1#av-roadmap)



(<https://datahack.analyticsvidhya.com/contest/data-science->

blogathon-8/?utm_source=Blog&utm_medium=stickybanner1)

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

Download App

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



(<https://apps.apple.com/us/app/analytics-vidhya/id1470025572>)

Analytics Vidhya

About Us (<https://www.analyticsvidhya.com/about-me/>)

Our Team (<https://www.analyticsvidhya.com/about-me/team/>)

Careers (<https://www.analyticsvidhya.com/about-me/career-analytics-vidhya/>)

Contact us (<https://www.analyticsvidhya.com/contact/>)

Data Science

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

Hackathon (<https://datahack.analyticsvidhya.com/>)

Discussions (<https://discuss.analyticsvidhya.com/>)

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

Companies

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

Trainings (<https://courses.analyticsvidhya.com/>)

Hiring Hackathons (<https://datahack.analyticsvidhya.com/>)

Advertising (<https://www.analyticsvidhya.com/contact/>)

Visit us**in**

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



(<https://www.youtube.com/channel/UCdH4hg3o2343iObA>)

© Copyright 2013-2020 Analytics Vidhya

[Privacy Policy](#) [Terms of Use](#) [Refund Policy](#)

