

PUBLISHED ON AUGUST 12, 2021

IN DEVELOPERS CORNER

([HTTPS://ANALYTICSINDIAMAG.COM/CATEGORY/DEVELOPERS-CORNER/](https://analyticsindiamag.com/category/developers-corner/))

Creating Deep Learning Models For Tabular Data using RTDL

RTDL (Revisiting Tabular Deep Learning) is an open-source Python package based on implementing the paper "Revisiting Deep Learning Models for Tabular Data". The library leverages the ease of creating a Deep Learning Model and can be used by practitioners and programmers looking to implement Deep Learning models in tabular data.

BY VICTOR ([HTTPS://ANALYTICSINDIAMAG.COM/AUTHOR/VICTOR-](https://analyticsindiamag.com/author/victor-)



The development of newer deep learning models in recent times and their ability to extract relevant information from different kinds of data has led to the creation of further possibilities to train models, algorithms to identify decisive patterns and discover crucial findings that general practitioners would not be able to discern. More

research in this field of data science has only recently started to appear. However, it has been getting lots of attention among the masses interested lately. The recent developments have led to delivering certain results that were not thought to be possible anytime before. Deep learning (<https://analyticsindiamag.com/hands-on-guide-to-differential-digital-signal-processing-using-neural-networks/>) can be defined as a machine learning technique that teaches computers to learn by example, just like humans do. For example, deep learning has been a key technology behind driverless-self driving cars, enabling them with the power and thinking to recognize a stop sign or to distinguish between a pedestrian and a lamppost. It is also the key to many voice control technologies in consumer devices like phones, tablets and smart TVs. In Deep Learning, the created computer model learns to perform classification tasks directly from images, text, or sound data.

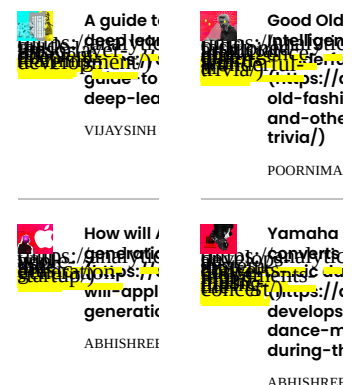
These models can help achieve state-of-the-art accuracy, sometimes even exceeding human-level performance. Models can be trained using a large set of labelled data, and its neural network architecture comprises several processing layers, which makes the network actually “Deep”. The larger the amount of labelled data, the more the recognition (<https://analyticsindiamag.com/face-recognition-system-using-deepface-with-python-codes/>) and classification (<https://analyticsindiamag.com/introduction-gans-generative-adversarial-networks/>) accuracy at higher levels. For example, creating a driverless car model would require millions of images and thousands of hours of videos to train the model on and help it understand better. Deep learning requires substantially high computing power as well. Well, structured models along with high-performance GPUs would make a more efficient deep learning architecture. When combined with clusters or cloud computing, this enables one to reduce the training time for a deep learning network to hours or less. Iterations within the model are continued until the output reaches an acceptable level of accuracy (<https://analyticsindiamag.com/hands-on-guide-to-padelpy-for-ml-model-building/>).

Deep learning techniques help eliminate some of the data pre-processing typically involved with traditional machine learning techniques. The input and output layers present in a deep neural network are known as visible layers. The input layer is a point where the deep learning model ingests the data from to process, and the output layer is where the final prediction or classification for the given problem is made. Real-world deep learning applications are a part of our daily lives these days. In most cases, they have been so well integrated into products and services that we users are at times unaware of the complex data processing that has been taking place in the background. Overall, using automatic feature engineering (<https://analyticsindiamag.com/visualizing-and-comparing-ml-models-using-lazypredict/>) and Deep Learning’s self-learning (<https://analyticsindiamag.com/exploring-panda-gym-a-multi-goal-reinforcement-learning-environment/>) capabilities, the algorithms need little to no human intervention. This also shows and tells us about the huge potential of Deep Learning and helps brainstorm and develop more ideas.



(<https://rising.analyticsindiasummit.>

More Stories



OUR UPCOMING EVENTS

- ▶ THE RISING 2022 | WOMEN IN AI CONFERENCE ()
- ▶ DATA ENGINEERING SUMMIT 2022 ()
- ▶ MACHINECON 2022 ()
- ▶ CYPHER 2022 ()

What is RTDL?

RTDL (Revisiting Tabular Deep Learning) is an open-source Python package based on implementing the paper “Revisiting Deep Learning Models for Tabular Data”. The library leverages the ease of creating a Deep Learning Model and can be used by practitioners and programmers looking to implement Deep Learning models in tabular data. It can also serve as a source of comparative baselines for researchers with other traditional libraries. Given the library’s high performance and simplicity, it can help for future works on tabular DL. It comprises a design based on Attention Transformer architectures.

Getting Started With Code

In this article, we will implement a basic Deep Learning Model using RTDL on tabular Data and predict the RMSE score that will provide us with the variability of the prediction accuracy and relative the variance of the model. The following implementation is partially inspired by the creators of RTDL, which can be accessed using the link [here \(https://yandex-research.github.io/rtdl/stable/rtdl.html\)](https://yandex-research.github.io/rtdl/stable/rtdl.html).

Installing the Library

To start with our model creation, we will first be installing the required libraries. The following lines can be run to do so,

```
#Installing required libraries
!pip install rtdl
!pip install libzero==0.0.4
```

We are also installing the libzero package here, which is a zero-overhead library for Pytorch.

Importing Dependencies

We will now be importing our further required dependencies for the RTDL library,

```
#importing the dependencies
import rtdl
import sklearn.datasets
import sklearn.model_selection
import sklearn.preprocessing
import torch
import torch.nn as nn
import torch.nn.functional as F
import zero
```

Loading and Processing the Data

Next, we will be loading our data to be processed. We will be using the California housing data set, readily available in sklearn, which contains housing data drawn from the 1990 U.S. Census. We will be splitting it into train and test and also preprocessing the features present.

```

#importing the dataset
dataset =
sklearn.datasets.fetch_california_housing()
X_all = dataset['data'].astype('float32')
y_all = dataset['target'].astype('float32')
X = {}
y = {}

#splitting into train test
X['train'], X['test'], y['train'], y['test'] =
sklearn.model_selection.train_test_split(
    X_all, y_all, train_size=0.8
)

#for validation
X['train'], X['val'], y['train'], y['val'] =
sklearn.model_selection.train_test_split(
    X['train'], y['train'], train_size=0.8
)

# Preprocess features present
preprocess =
sklearn.preprocessing.StandardScaler().fit(X['train'])

X = {
    k: torch.tensor(preprocess.fit_transform(v),
device=device)
    for k, v in X.items()
}

# applying formula for solving regression problem
y_mean = float(y['train'].mean())
y_std = float(y['train'].std())
y = {
    k: torch.tensor((v - y_mean) / y_std,
device=device)
    for k, v in y.items()
}

```

We will be applying a feature transformer, which will help improve the model's performance by reducing bias, defining relationships, removing outliers, and more.

```

#Applying Feature Transformer
model = rtdl.FTTransformer.make_default(
    n_num_features=X_all.shape[1],
    cat_cardinalities=None,
    last_layer_query_idx=[-1],
    d_out=1,
)

#setting up the optimizer model

model.to(device)
optimizer = (
    model.make_default_optimizer()
    if isinstance(model, rtdl.FTTransformer)
    else torch.optim.AdamW(model.parameters()),
    lr=lr, weight_decay=weight_decay)
)

```

Final Model Setup

Finally, let us set up the model pipeline that will help us apply the feature transformer and metrics to calculate the RMSE value.

```

#applying the model
def apply_model(x_num, x_cat=None):
    # rtdl.FTTransformer expects two inputs: x_num
    and x_cat
    return model(x_num, x_cat) if
isinstance(model, rtdl.FTTransformer) else
model(x_num)

@torch.no_grad()
def evaluate(part):
    model.eval()

#calculating rmse
mse =
F.mse_loss(apply_model(X[part])).squeeze(1),
y[part]).item()
rmse = mse ** 0.5 * y_std
return rmse

#Setting the batch size
batch_size = 256
train_loader =
zero.data.IndexLoader(len(X['train']), batch_size,
device=device)

progress = zero.ProgressTracker(patience=100)

print(f 'Test RMSE before training:
{evaluate("test"):.4f}')

```

Calculating RMSE & Best Validation Epoch

We derive our final output and test the model metrics while printing the best validation epoch value to see how good our model can perform.

```
#setting epoch size
n_epochs = 50
for epoch in range(1, n_epochs + 1):
    for batch_idx in train_loader:
        model.train()
        optimizer.zero_grad()
        x_batch = X['train'][batch_idx]
        y_batch = y['train'][batch_idx]

        F.mse_loss(apply_model(x_batch).squeeze(1),
y_batch).backward()
        optimizer.step()

        val_rmse = evaluate('val')
        test_rmse = evaluate('test')
        print(f'Epoch {epoch:03d} | Validation RMSE:
{val_rmse:.4f} | Test RMSE: {test_rmse:.4f}',
end='')
        progress.update(-val_rmse)
        if progress.success:
            print(' <<< BEST VALIDATION EPOCH',
end='')
            print()
            if progress.fail:
                break
```

Output :

Epoch 001 | Validation RMSE: 0.8441 | Test RMSE: 0.5852
Epoch 002 | Validation RMSE: 0.8252 | Test RMSE: 0.5731
Epoch 003 | Validation RMSE: 0.8061 | Test RMSE: 0.5749 <<< BEST VALIDATION EPOCH
Epoch 004 | Validation RMSE: 0.7962 | Test RMSE: 0.5685 <<< BEST VALIDATION EPOCH
Epoch 005 | Validation RMSE: 0.8068 | Test RMSE: 0.5722
Epoch 006 | Validation RMSE: 0.8254 | Test RMSE: 0.5689
Epoch 007 | Validation RMSE: 0.7939 | Test RMSE: 0.5705 <<< BEST VALIDATION EPOCH
Epoch 008 | Validation RMSE: 0.7914 | Test RMSE: 0.5724 <<< BEST VALIDATION EPOCH
Epoch 009 | Validation RMSE: 0.8042 | Test RMSE: 0.5596
Epoch 010 | Validation RMSE: 0.8531 | Test RMSE: 0.5759
Epoch 011 | Validation RMSE: 0.7862 | Test RMSE: 0.5678 <<< BEST VALIDATION EPOCH
Epoch 012 | Validation RMSE: 0.7942 | Test RMSE: 0.5651
Epoch 013 | Validation RMSE: 0.7868 | Test RMSE: 0.5715
Epoch 014 | Validation RMSE: 0.7719 | Test RMSE: 0.5812 <<< BEST VALIDATION EPOCH
Epoch 015 | Validation RMSE: 0.7440 | Test RMSE: 0.5695 <<< BEST VALIDATION EPOCH
Epoch 016 | Validation RMSE: 0.7833 | Test RMSE: 0.5657
Epoch 017 | Validation RMSE: 0.8052 | Test RMSE: 0.5711
Epoch 018 | Validation RMSE: 0.7634 | Test RMSE: 0.5750
Epoch 019 | Validation RMSE: 0.7330 | Test RMSE: 0.5661 <<< BEST VALIDATION EPOCH
Epoch 020 | Validation RMSE: 0.7520 | Test RMSE: 0.5582
Epoch 021 | Validation RMSE: 0.8038 | Test RMSE: 0.5611
Epoch 022 | Validation RMSE: 0.7813 | Test RMSE: 0.5636
Epoch 023 | Validation RMSE: 0.7614 | Test RMSE: 0.5764
Epoch 024 | Validation RMSE: 0.7748 | Test RMSE: 0.5704
Epoch 025 | Validation RMSE: 0.7430 | Test RMSE: 0.5589
Epoch 026 | Validation RMSE: 0.7686 | Test RMSE: 0.5487
Epoch 027 | Validation RMSE: 0.7350 | Test RMSE: 0.5523
Epoch 028 | Validation RMSE: 0.7862 | Test RMSE: 0.5596

```
Epoch 029 | Validation RMSE: 0.7472 | Test RMSE: 0.5727
Epoch 030 | Validation RMSE: 0.7427 | Test RMSE: 0.5603
Epoch 031 | Validation RMSE: 0.7618 | Test RMSE: 0.5583
Epoch 032 | Validation RMSE: 0.7394 | Test RMSE: 0.5573
Epoch 033 | Validation RMSE: 0.7671 | Test RMSE: 0.5607
Epoch 034 | Validation RMSE: 0.7604 | Test RMSE: 0.5633
Epoch 035 | Validation RMSE: 0.7439 | Test RMSE: 0.5540
Epoch 036 | Validation RMSE: 0.7596 | Test RMSE: 0.5533
Epoch 037 | Validation RMSE: 0.7731 | Test RMSE: 0.5621
Epoch 038 | Validation RMSE: 0.7589 | Test RMSE: 0.5584
Epoch 039 | Validation RMSE: 0.7883 | Test RMSE: 0.5617
Epoch 040 | Validation RMSE: 0.7690 | Test RMSE: 0.5644
Epoch 041 | Validation RMSE: 0.7461 | Test RMSE: 0.5623
Epoch 042 | Validation RMSE: 0.7671 | Test RMSE: 0.5659
Epoch 043 | Validation RMSE: 0.7668 | Test RMSE: 0.5668
Epoch 044 | Validation RMSE: 0.7702 | Test RMSE: 0.5544
Epoch 045 | Validation RMSE: 0.7772 | Test RMSE: 0.5570
Epoch 046 | Validation RMSE: 0.7692 | Test RMSE: 0.5698
Epoch 047 | Validation RMSE: 0.7707 | Test RMSE: 0.5696
Epoch 048 | Validation RMSE: 0.7631 | Test RMSE: 0.5704
Epoch 049 | Validation RMSE: 0.7253 | Test RMSE: 0.5638 <<< BEST VALIDATION EPOCH
Epoch 050 | Validation RMSE: 0.7693 | Test RMSE: 0.5623
```

End Notes

In this article, we understood what Deep Learning models are and their importance, also discussed how these models and algorithms are being used. We also explored the RTDL library and implemented a basic Deep Learning model for tabular data with it. The following implementation above can be found as a Colab notebook and accessed using the link [here](https://colab.research.google.com/drive/1zPyQiEPLoVOsz9bRdJb-NvHx5Escs4_3?usp=sharing) (https://colab.research.google.com/drive/1zPyQiEPLoVOsz9bRdJb-NvHx5Escs4_3?usp=sharing).

Happy Learning!

References

- [Official RTDL Paper \(https://arxiv.org/pdf/2106.11959.pdf\)](https://arxiv.org/pdf/2106.11959.pdf)
- [RTDL Github Repository \(https://github.com/yandex-research/rtdl#2-overview\)](https://github.com/yandex-research/rtdl#2-overview)

More Great AIM Stories

[The First Quantum Toolkit And For Natural Language Processing \(https://analyticsindiamag.com/natural-language-processing/\)](https://analyticsindiamag.com/natural-language-processing/)

[NVIDIA's Large Language AI Models Are Now Available To Businesses Worldwide \(https://analyticsindiamag.com/language-ai-models/\)](https://analyticsindiamag.com/language-ai-models/)

[A Tutorial on Spiking Neural Networks for Beginners \(https://analyticsindiamag.com/tutorial-on-spiking-neural-networks-for-beginners/\)](https://analyticsindiamag.com/tutorial-on-spiking-neural-networks-for-beginners/)

[Stanford Brings Out BEHAVIOR Benchmark For 100 Everyday Household Tasks \(https://analyticsindiamag.com/brings-out-behavior-benchmark-100-everyday-household-tasks/\)](https://analyticsindiamag.com/brings-out-behavior-benchmark-100-everyday-household-tasks/)

[Meta AI Puts A Step Towards Building Universal Translation System \(https://analyticsindiamag.com/ai-puts-a-step-towards-building-universal-translation-system/\)](https://analyticsindiamag.com/ai-puts-a-step-towards-building-universal-translation-system/)

[A Complete Tutorial on Expert Systems \(https://analyticsindiamag.com/complete-tutorial-on-expert-systems/\)](https://analyticsindiamag.com/complete-tutorial-on-expert-systems/)



[\(https://analyticsindiamag.com/author/victor-deyanalyticsindiamag-com/\)](https://analyticsindiamag.com/author/victor-deyanalyticsindiamag-com/)

Victor is an aspiring Data Scientist & is a Master of Science in Data Science & Big Data Analytics. He is a Researcher, a Data Science Influencer and also an Ex-University Football Player. A keen learner of new developments in Data Science and Artificial Intelligence, he is committed to growing the Data Science community.

Post your Data Science Queries

OVER THE LAST 10 YEARS, WE HAVE BUILT THE BIGGEST DATA SCIENCE COMMUNITY IN INDIA.

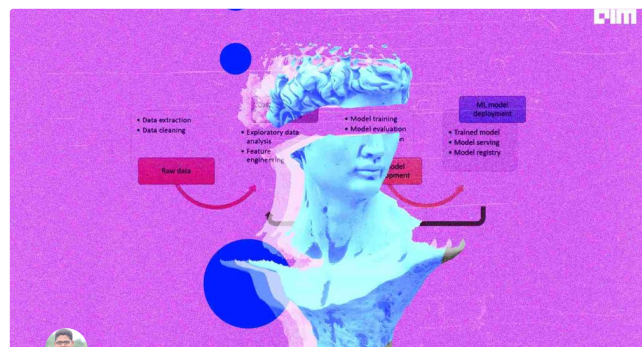
POST QUESTION
([HTTPS://MACHINEHACK.COM/DISCUSSION](https://machinehack.com/discussion))

MORE FROM AIM



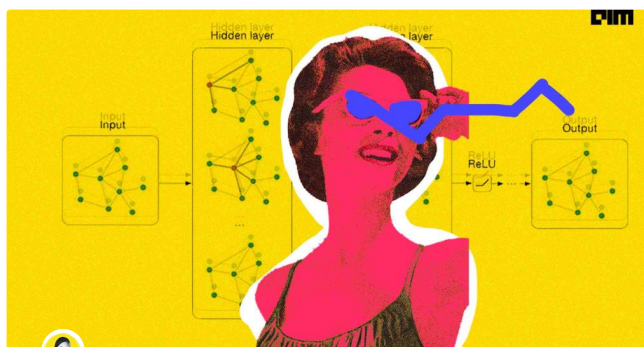
A guide to TensorLayer for efficient deep learning development (<https://analyticsindiamag.com/a-guide-to-tensortlayer-for-efficient-deep-learning-development/>)

In this post, we'll look at TensorLayer, a Python-based machine learning tool.



Step by step guide to deploy an ML model using Heroku (<https://analyticsindiamag.com/step-by-step-guide-to-deploy-an-ml-model-using-heroku/>)

In this article, we will discuss cloud-based deployment using the Heroku cloud service.



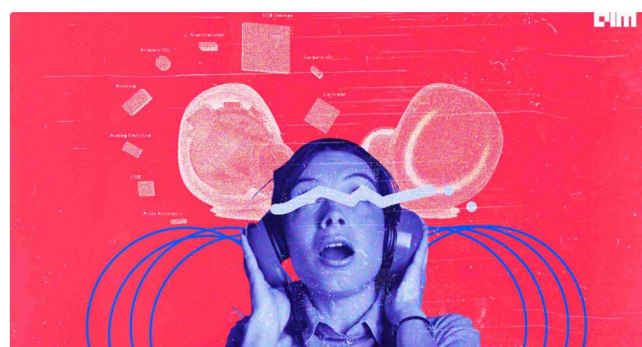
Top resources to learn graph neural networks (<https://analyticsindiamag.com/top-resources-to-learn-graph-neural-networks/>)

Let us take a look at a few of the courses and tools to get you started with understanding more about GNNs.



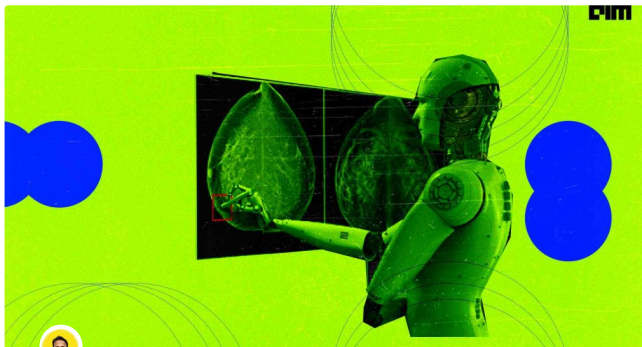
Meet the winners of the 'Dare In Reality' hackathon (<https://analyticsindiamag.com/meet-the-winners-of-the-dare-in-reality-hackathon/>)

Let's take a look at the winners of the 'Dare in Reality' hackathon who took home cash prizes and goodies.



Microsoft, NVIDIA test waters for a large-scale generative language model with promising results (<https://analyticsindiamag.com/microsoft-nvidia-test-waters-for-a-large-scale-generative-language-model-with-promising-results/>)

We believe that our results and findings can help, shape, and facilitate future research in foundational, large-scale pretraining.



Chinese university develops AI tool to detect early-stage lung cancer (<https://analyticsindiamag.com/chinese-university-develops-ai-tool-to-detect-early-stage-lung-cancer/>)

The detector achieved over 90% accuracy.

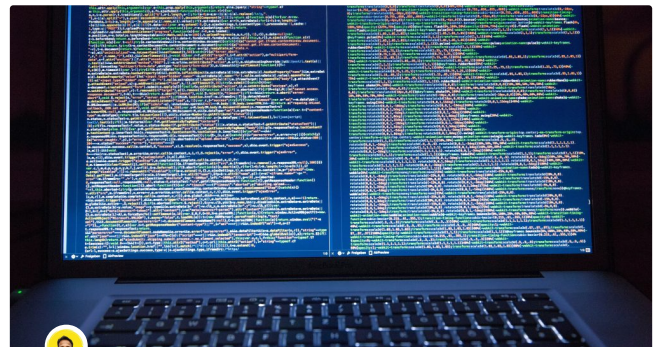


A beginner's guide to image processing using NumPy (<https://analyticsindiamag.com/a-beginners-guide-to-image-processing-using-numpy/>)

Since images can also be considered as made up of arrays, we can use NumPy for performing different image processing tasks as well from scratch. In this article, we will learn about the image processing tasks that can be performed only using NumPy.

Top AI-Powered hearables that must be on your wishlist in 2022! (<https://analyticsindiamag.com/top-ai-powered-hearables-that-must-be-on-your-wishlist-in-2022-%ef%bf%bc/>)

Artificial Intelligent Noise Cancellation technology continuously analyses ambient sound components to select the most effective noise cancelling filter.



A new programming language called ATL for high-performance computing (<https://analyticsindiamag.com/a-new-programming-language-called-atl-for-high-performance-computing/>)

ATL is the only tensor language with formally verified optimisations that has been tested on a number of small programs.



Copilot vs AlphaCode: The race for coding supremacy (<https://analyticsindiamag.com/copilot-vs-alphacode-the-race-for-coding-supremacy/>)

Deepmind's AlphaCode made headlines by testing in the top 54% of human coders. Can GitHub's Copilot keep up with AlphaCode's automated programming?

3 Ways to Join our Community

Discord Server

Stay Connected with a larger ecosystem of data science and ML Professionals

JOIN DISCORD COMMUNITY
([HTTPS://DISCORD.GG/SBTJ3JDEAZ](https://discord.gg/SBTJ3JDEAZ))

Telegram Channel

Discover special offers, top stories, upcoming events, and more.

JOIN TELEGRAM
([HTTPS://T.ME/+TRPAPV7GNN2OZ1AZ](https://t.me/+TRPAPV7GNN2OZ1AZ))

Subscribe to our newsletter

Get the latest updates from AIM

OUR MISSION IS TO BRING ABOUT BETTER-INFORMED AND MORE CONSCIOUS DECISIONS ABOUT TECHNOLOGY THROUGH AUTHORITATIVE, INFLUENTIAL, AND TRUSTWORTHY JOURNALISM.

SHAPE THE FUTURE OF AI

CONTACT US →

([HTTPS://ANALYTICSINDIAMAG.MYSTAGINGWEBSITE.COM/CONTACT-US/](https://analyticsindiamag.mystagingwebsite.com/contact-us/))



(<https://analyticsindiamag.com>)

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

[About Us](#)
[Advertise](#)
[Weekly Newsletter](#)
[Write for us](#)
[Careers](#)
[Contact Us](#)

RANKINGS & LISTS

[Academic Rankings](#)
[Best Firms To Work For](#)
[Top Leaders](#)
[Emerging Startups](#)
[Trends](#)
[PeMa Quadrant](#)

OUR BRANDS

[AIM Research](#)
[AIM Recruits](#)
[AIM Leaders Council](#)

VIDEOS

[Documentary – The Transition Cost](#)
[Web Series – The Dating Scientists](#)
[Podcasts – Simulated Reality](#)
[Analytics India Guru](#)
[The Pretentious Geek](#)
[Deeper Insights with Leaders](#)
[Curiosum – AI Storytelling](#)

AWARDS

- Analytics100
- 40 under 40 Data Scientists
- Women in AI Leadership
- Data Science Excellence

EVENTS

- AIM Custom Events
- AIM Virtual

MACHINEHACK

- For Organizations
- Hackathons
- Discussion Forum
- Job Portal
- Mock Assessments
- Practice ML
- Courses

OUR CONFERENCES

- Cypher
- The MachineCon
- Machine Learning Developers Summit
- The Rising
- Data Engineering Summit

NEWSLETTER

Stay up to date with our latest news, receive exclusive deals, and more.

Enter Your Email Address

SUBSCRIBE →