

Installing And Setting Up UrBackup On Ubuntu

Testing SaltStack Environment With KitchenSalt, Docker And TestInfra

₹120
ISSN-2456-4885

OpenSource For You

Volume: 11 | Issue: 02 | Pages: 100 | December 2022

THE COMPLETE MAGAZINE ON OPEN SOURCE

An EFY GROUP Publication

Great Ways To Manage Data



Open Source Tools For Data Monetisation: A Quick Look

RAWGraphs: The No-Code Data Visualisation Tool

Using Apache Beam To Pipeline And Process Data



Make A Calories Burn Prediction On A Fitness Tracker Data Set



The Connect Between Open Source And The Metaverse

www.opensourceforu.com

Wanna Be Your Own Boss?

DO OPEN SOURCE.



Demand for Open Source is sky rocketing. Be it for managing IT infrastructure or development of software--Open Source solutions are what customers are seeking.

All you need to do is develop expertise in an Open Source stack, and then build a team around it!

And, Open Source For You can be your friend and a guide through this journey.

TO READ OUR PRINT EDITION Visit: https://subscribe.efyindia.com

TO READ OUR EZINE EDITION Visit: https://ezine.lfymag.com



WORLD'S LEADING PUBLICATION ON OPEN SOURCE

Looking for marketing solutions to engage with cutting edge techies?
Contact us at growmybiz@efy.in OR call us at +91-9811155335.

Does Your Antivirus Solution Provide You With Complete Protection?

Don't RELY on 20 years old technology to FIGHT current date viruses



TOP 10 things that your Antivirus solution should provide to tackle today's threat...

Ransomware File Protection	ATP- Advance Threat Protection	EDR- Endpoint Detection And Response	
Application Blocking	Deep Learning Malware Analysis	Exploit Prevention	URL Blocking
Disk and Boot Record Protection	Peripheral Control	Respond Investigate Remove (Root Cause Analysis)	

If it doesn't, contact us for a solution which can

Contact: Santosh on 9971696319 or Email at santosh.gupta@itsipl.com

ITS
A Total Solutions Company

I. T. Solutions India Private Limited
D-88/5, Okhla Industrial Area, Okhla Phase I, New Delhi -110020
Ph: 011-47695000 • Email: sales@ itsipl.com • www.itsipl.com

- Mumbai
- Jaipur
- Chandigarh

SOPHOS 2020 BEST NEXTGEN PARTNER | Delhi/ NCR
SOPHOS 2019 BEST NEXTGEN PARTNER | Delhi/ NCR
SOPHOS 2018 BEST SI PARTNER | Delhi

CONTENTS

DECEMBER 2022 | ISSN-2456-4885

FOR U & ME

14 State of Open Source: Trends, Insights and Lessons

18 The Evolution of IoT Architecture

20 Innovative Lessons from Ladakh for Rest of the World

25 Why Private 5G Services are Needed for Implementing Industry 4.0

30 The Connect Between Open Source and the Metaverse

FOCUS

46 Installing and Setting Up UrBackup on Ubuntu

DEVELOPERS

53 How to Make a Calories Burn Prediction on a Fitness Tracker Data Set

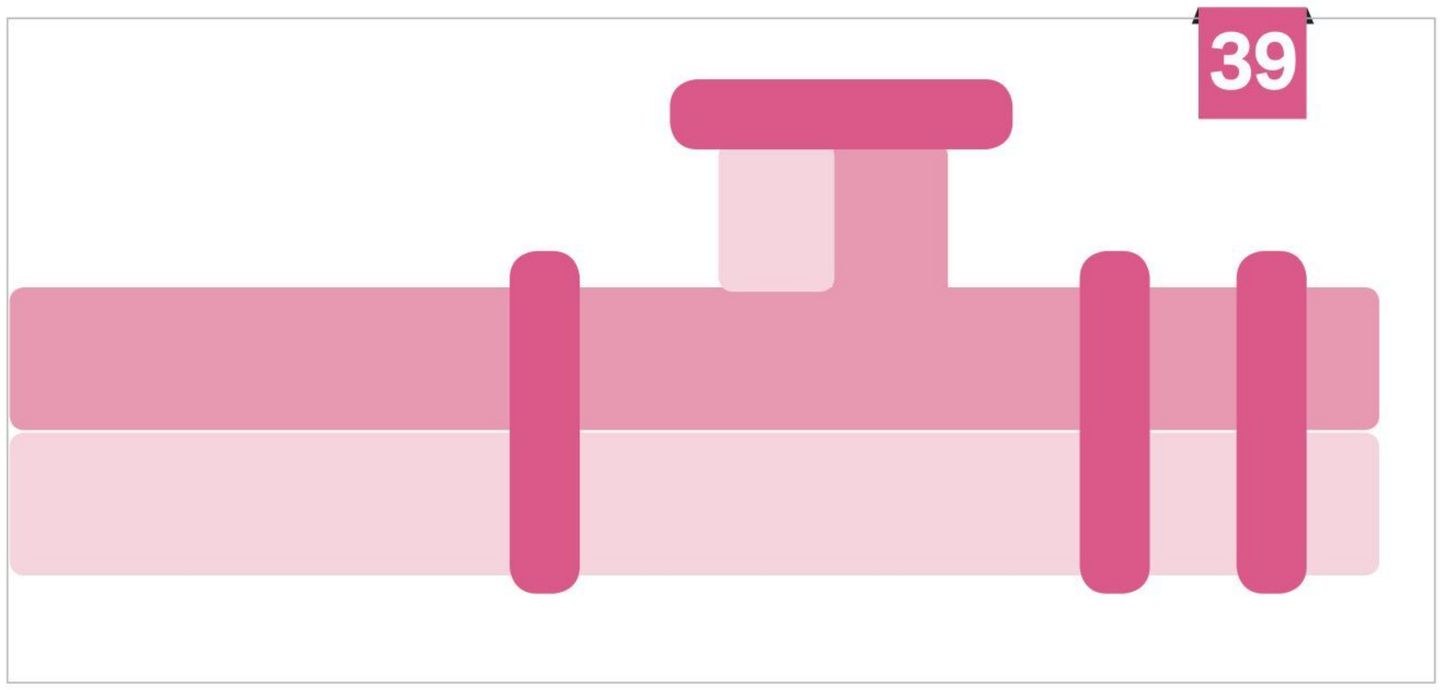
60 Onboard IIoT Devices Seamlessly with the FDO Standard

68 R Series: Correlation

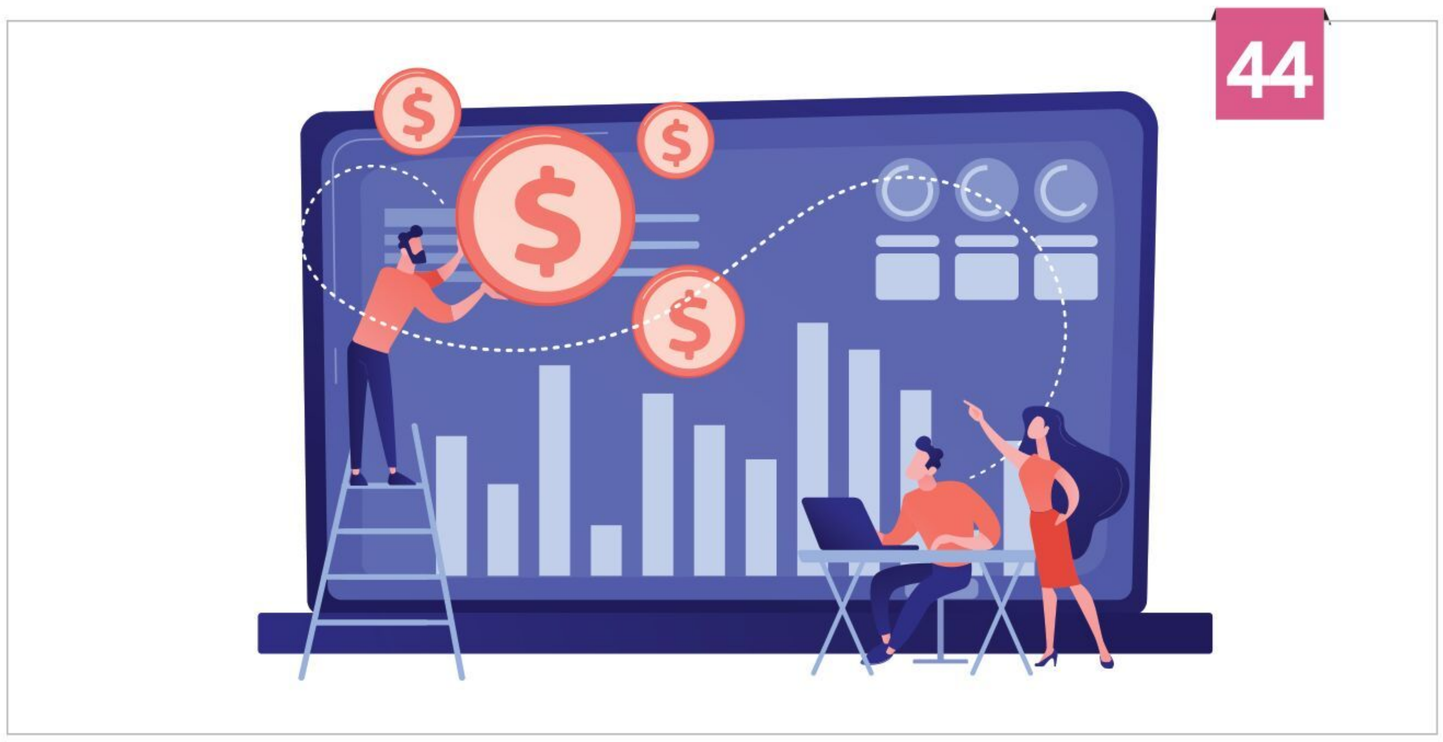
OPEN GURUS

88 Connecting Arduino to the Internet

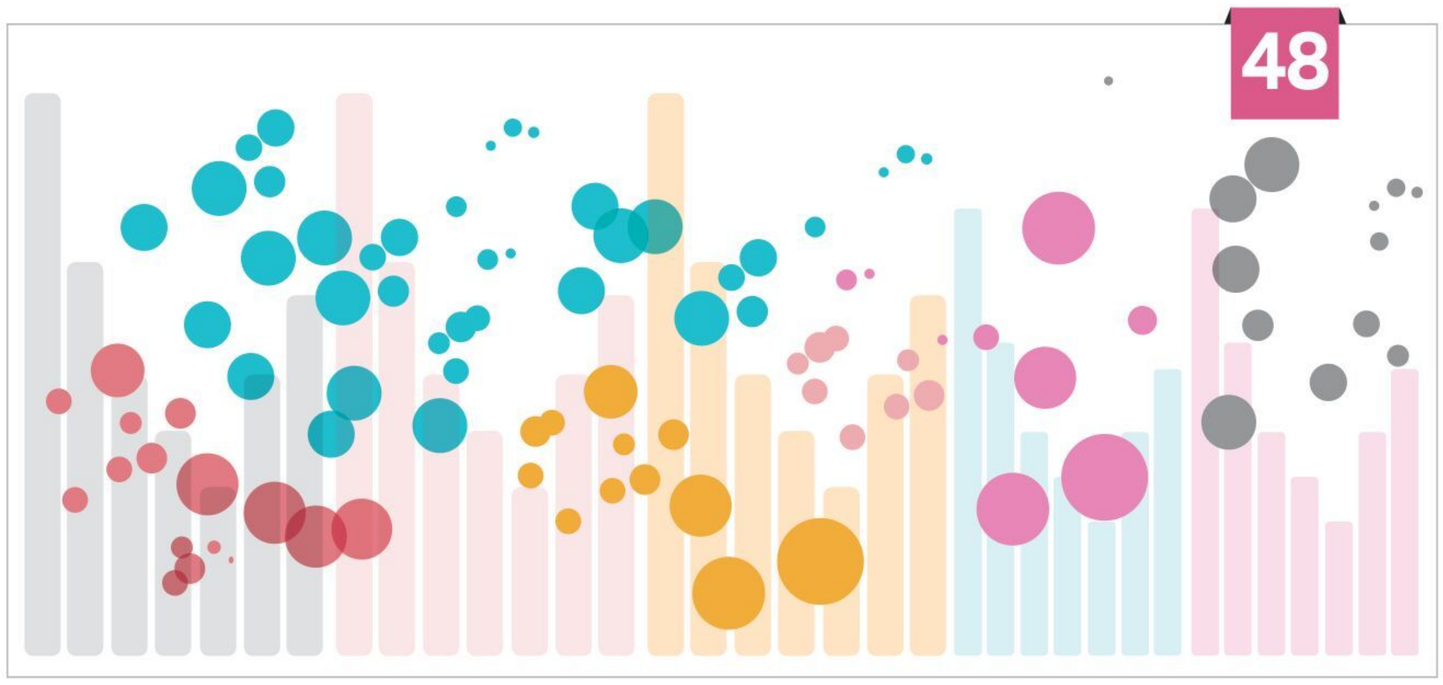
92 A Robot that can Dream, Recognise and Greet Visitors



Using Apache Beam to Pipeline and Process Data



Open Source Tools for Data Monetization: A Quick Look



RAWGraphs: The No-Code Data Visualisation Tool

REGULAR FEATURES

07 FossBytes

PIONEERS IN TECHNOLOGY.
PARTNERS IN SUCCESS.

EFY GROUP
YOURS SINCE 1969

"It has helped me reach this position."

-Mukesh Ajwani
MD, Electronic Components
Trading Firm



Mr Ajwani is not the only one who owes his success to *Electronics For You*. The technology magazine that started off in 1969 is today, South Asia's most popular one. Covering the latest on emerging technologies, changing industry trends, evolving trade practices...

Not surprisingly, top decision makers and those high up on the corporate ladder in the electronics fraternity treat EFY as their Bible.

- **Over half a million readers**
- **India's largest selling technology magazine**
- **Caters to the entire electronics fraternity**

D-87/1, Okhla Industrial Area, Phase 1, New Delhi-20 • Ph.: 011-26810601/02/03 • Email: info@efy.in • www.electronicsforu.com

EDITOR
RAHUL CHOPRA

EDITORIAL, SUBSCRIPTIONS & ADVERTISING
Delhi (HQ)
D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020
Phone: +91-9811155335
E-mail: info@efy.in

MISSING ISSUES
Phone: +91-9811155335
E-mail: support@efy.in

BACK ISSUES
Phone: +91-9811155335
E-mail: support@efy.in

NEWSSTAND DISTRIBUTION
Phone: +91-9811155335
E-mail: efycirc@efy.in

ADVERTISEMENTS
NEW DELHI (HEAD OFFICE)
Phone: +91-9811155335
E-mail: efyenq@efy.in

MUMBAI
E-mail: rmowest@efy.in

BENGALURU
E-mail: rmosouth@efy.in

CHINA
Worldwide Focus Media
E-mail: china@efy.in

GERMANY
pms Plantenberg Media Service GmbH
E-mail: germany@efy.in

JAPAN
Tandem Inc.
E-mail: japan@efy.in

TAIWAN
J.K. Media
E-mail: taiwan@efy.in

UNITED KINGDOM
ASA Media
E-mail: uk@efy.in

UNITED STATES
E & Tech Media
E-mail: usa@efy.in

Printed, published and owned by Ramesh Chopra. Printed at Tara Art Printers Pvt Ltd, A-46,47, Sec-5, Noida, on 28th of the previous month, and published from D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020. Copyright © 2021. All articles in this issue, except for interviews, verbatim quotes, or unless otherwise explicitly mentioned, will be released under Creative Commons Attribution-NonCommercial 3.0 Unported License a month after the date of publication. Refer to http://creativecommons.org/licenses/by-nc/3.0/ for a copy of the licence. Although every effort is made to ensure accuracy, no responsibility whatsoever is taken for any loss due to publishing errors. Articles that cannot be used are returned to the authors if accompanied by a self-addressed and sufficiently stamped envelope. But no responsibility is taken for any loss or delay in returning the material. Disputes, if any, will be settled in a New Delhi court only.

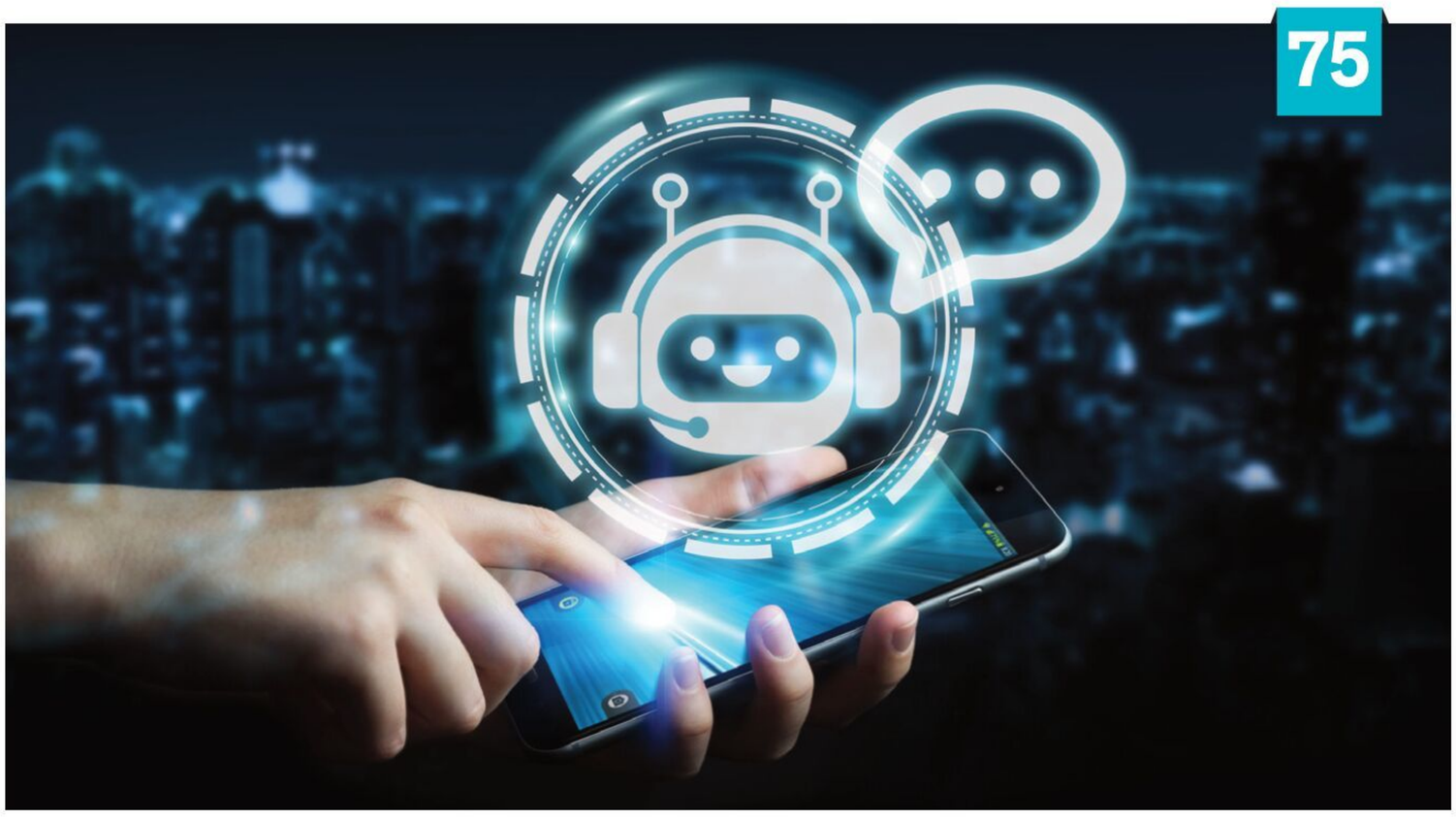
SUBSCRIPTION RATES			
Year	**Newstand Price (₹)**	**You Pay (₹)**	**Overseas**
Five	7200	4320	—
Three	4320	3030	—
One	1440	1150	US$ 120

Kindly add ₹ 50/- for outside Delhi cheques.
Please send payments only in favour of EFY Enterprises Pvt Ltd.
Non-receipt of copies may be reported to **support@efy.in**—do mention your subscription number.

CONTENTS



AI: Introduction to Keras and Our First Data Set



It's the Era of Conversational AI



Testing SaltStack Environment with KitchenSalt, Docker and TestInfra



Setting Up an Email Server with Carbonio CE

Shell signs MoU to help decarbonise businesses in China



Shell has signed a memorandum of understanding (MoU) with Sinopec, Baowu and BASF to find out the possibility of developing an open source carbon capture, utilisation and storage (CCUS) project in the East China region.

"The four parties intend to conduct a joint study to assess the technical solutions and develop a commercial model for the project. The study will also explore to establish high-integrity and verified low-carbon product supply chains and propose enabling policies. If successful, it will be China's first large-scale open-source CCUS project with a potential capacity of tens of million tonnes of CO^2 per year," said a press release from Shell.

Shell has mooted that an open source project may give industrial firms situated near the middle and lower banks of the Yangtze River chances to capture and store their CO^2 emissions. If this idea is effective, CO^2 emissions from industrial businesses in the East China region may be captured too. These emissions may then be delivered to a receiving terminal via CO^2 carrier ships, where they would subsequently be transported via brief pipelines to onshore and offshore storage facilities. According to Shell, this might present a flexible, effective, and integrated decarbonisation option for businesses in the area.

With an estimated 2,400 gigatonnes of storage capacity, China has significant geological potential for storing carbon, second only to the USA.

Gitpod raises US$ 25 million in funding

Gitpod GmbH, a startup open source developer platform, has raised US$ 25 million in fresh funding to establish 'cloud development environments'.

The Series A investment was led by Tom Preston-Werner, the creator and former CEO of GitHub. Other notable individual investors include General Catalyst, Crane Venture Partners, Vertex Ventures US, Speedinvest, Pebblebed, GTMfund, and MongoDB Ventures. According to information from Crunchbase, Gitpod has now raised US$ 14 million in total, including the fresh capital.

Gitpod, a 2019 startup, provides an online integrated development environment that can be opened from any GitHub page in a web browser. It offers a fully functional programming environment with support for desktop or browser-based VS

Matt Klein, Envoy's creator, is busy 'plumbing' the system

Envoy, an open source edge and service proxy for cloud-native applications, was developed by Matt Klein, a software engineer at Lyft, in 2015, in order to bridge the gap between various application frameworks and programming languages. In 2017, he contributed it to the Cloud Native Computing Foundation. As a result of its widespread use, Klein now devotes a large portion of his time to maintaining the Envoy's plumbing, and calls himself a 'plumber' on his Twitter handle.

At KubeCon + CloudNativeCon NA 2022, Klein discussed the industry with analysts John Furrier and Savannah Peterson in an exclusive interview that was televised on theCUBE, SiliconANGLE Media's livestreaming studio. They talked about the most recent Envoy version and new developer-oriented open source tools.

Envoy Gateway, which aims to increase accessibility for API platforms that wish to work with cloud-native tools like service meshes and Kubernetes, was added to the edge proxy in June this year. "The goal of Envoy Gateway is to make it easier for people to run Envoy within Kubernetes as an ingress controller," Klein said. "We want to provide Envoy Gateway as a way of onboarding more users into the Envoy ecosystem and making Envoy the default API gateway or edge proxy within Kubernetes."

Grafana Labs announces two new open source tools

Grafana Labs Inc. has made available two open source tools that will assist developers in streamlining their applications and fixing bugs more quickly. The New York based company is funded to the tune of more than US$ 535 million by Sequoia Capital, Coatue, and other investors. One of the most well-liked open source solutions for spotting technical problems in IT infrastructure, Grafana, was created by the company.

Grafana Phlare is the first new open source product that Grafana Labs has introduced. It enables programmers to monitor CPU usage, memory needs, and other hardware consumption parameters of an application. Grafana claims that the data supplied by Phlare can assist developers in locating chances to improve the resource-efficiency of their product.

The second open source product that has been unveiled is Grafana Faro. It will assist developers in identifying any technological problems that can harm an application's user interface. The technology gathers technical information regarding conceivable flaws and makes it accessible via a centralised monitoring dashboard.



Developers typically need to add a sizable amount of so-called instrumentation code to an application's code base in order to gather information about faults in the application. According to Grafana, Faro makes the procedure simpler. The startup claims that two lines of code allow the tool to be deployed in a matter of seconds.

Code or any JetBrains integrated development environment, enabling developers to start coding in only a few seconds. The exact project that is being updated is automatically configured in a separate cloud based Linux container.



Gitpod's platform was being used by more than 350,000 developers as of its most recent investment round in April 2021. This figure has gone up to 750,000 people. Gitpod is used by developer teams at notable organisations like Google LLC, GitLab Inc., DataStax Inc., and Amazon Web Services Inc.

The organisation guarantees instantaneous accessibility and offers secure development settings. A workspace plugin system, application programming interfaces, and enhanced extensibility are on the development agenda.

Dyndrite is now compatible with the new open vector format

Dyndrite has announced compatibility of its Dyndrite Laser Powder Bed Fusion (LPBF) software with the open vector format (OVF) created by the RWTH Aachen University Chair Digital Additive Production DAP. Scientists at RWTH Aachen University created the open source OVF data format to offer a simplified, information-rich data interface between digital and physical processes. This enables a solid and effective manufacturing process, and also has "a number of advantages over existing formats like CLI and 3MF."

With the help of a wide range of tools that cover file conversion to OVF files, integrity check routines, and contour, parameter, and layer checks, LPBF process-relevant metadata about manufacturing parameters, laser powder, and scanning speed can be efficiently transported along the process chain with OVF. These tools are available on the OVF GitHub.

The lack of a standardised format for the output data of LPBF processes, according to Dyndrite, is a challenge that has been solved by RWTH Aachen University's creation of OVF. Dyndrite has integrated OVF export functionality directly into its Application Development Kit (ADK) and LPBF products through a partnership with RWTH Aachen University.

"We developed OVF because there was no satisfactory format for 2.5-dimensional data that was both open-readable and contained additional information besides the actual toolpath," said Moritz Kolter, group manager digital production, RWTH Aachen University Chair Digital Additive Production DAP. "OVF, however, is able to process data after slicing in a performant, readable

manner and is also able to link other information such as 3D part data or metadata. This is especially important in order to have a fully linked end-to-end data chain that will drive the industrialisation of AM. For example, we use it for advanced production planning or part orientation algorithms to further promote the sustainable and efficient use of technologies such as LPBF."

KGS releases GIRYPTO, a new open source IP for Web5

With the beginning of its multi-phase GIRYPTO NFT project, Malaysian comic book publisher Kadokawa Gempak Starz (KGS) is launching a brand new open source intellectual property (IP) for Web5 (Web3 + Web2) named GIRYPTO. KGS is bringing value to the metaverse with the launch of this IP as it forays into the blockchain industry.



GIRYPTO is the first open source IP in the comic book industry, which is what makes it so special. Anyone and everyone is free to copy and distribute GIRYPTO contents in any format or medium, as well as remix, adapt, and build upon them for any purpose, even commercially, under the terms of the CC BY-SA 4.0 licence agreement, as long as proper acknowledgment is given.

The GIRYPTO NFT project is the first of many GIRYPTO projects that will be introduced. The GIRYPTO Fables NFT series was made available to the public on November 11, 2022. Chibi (cute) illustrations of GIRYPTO characters are included in this NFT series. There are 3,000 pieces available of this free-to-mint series. The Fables series also serves as a mint pass for the GIRYPTO Epics NFT series, scheduled to debut in December 2022.

The Epics series, on the other hand, only has 999 pieces and shows the characters in their true form. Fans are advised to keep up with the news as KGS is also working with renowned comic creators from Japan's Kadokawa Corporation to develop crossovers that will be featured in some of its GIRYPTO NFTs.

S3crets Scanner searches for 'secrets' in AWS S3 buckets

Researchers and red-teamers can look for 'secrets' accidentally saved in publicly accessible or company-owned Amazon AWS S3 storage buckets using the new open source 'S3crets Scanner'. Companies frequently utilise Amazon S3 (Simple Storage Service), a cloud storage service, to store software, services, and data in bucket-like containers.

Viasat and cyan partner to increase connectivity services globally

Viasat, Inc., a global communications company, has signed an agreement leveraging cyan's BSS/OSS platform to facilitate the expansion of global connectivity and communication services. cyan AG is a European and globally active supplier of BSS/OSS platform services and intelligent cybersecurity solutions for the telecom industry.

The BSS/OSS platform from cyan provides communication service providers with a one-stop-shop solution with flexibility and scalability. Real-time charging, network integration, product administration, customer service, and digital customer experience management are all supported by the architecture.

More Viasat end users can benefit from its satellite connectivity in challenging-to-reach regions around the world thanks to cyan's BSS/OSS platform. It is anticipated that Viasat's satellite constellation, ViaSat-3, will be able to supply even more bandwidth, leading to an increase in speed, data, and streaming choices.



"We are entering a new phase for our digital BSS/OSS platform together with Viasat. Viasat is our worldwide satellite provider we welcome as customer, and we are confident that others will follow as rural areas in almost all continents will have a higher demand to bring connectivity solutions to every person around the globe," said Frank von Seth, CEO of cyan.

ActiveState will help software vendors comply with White House mandate

ActiveState is the first open source software platform to offer the crucial element of software supply chain security with the announcement of the availability of open source software attestations. A comprehensive set of recommendations from the National Institute of Standards and Technology (NIST) that has been accepted by the White House includes the ability to receive self-attestation for all third-party software and the capacity to create a software bill of materials (SBOM). A recent White House directive mandates that by June 12, 2023, all essential software that interacts in any way with government data or systems must meet these new security requirements. By September 14, 2023, all software must abide by these stringent requirements.

Most firms do not carefully maintain open source provenance. While some very large organisations may have the systems and procedures in place to comply with the mandate, this will be a costly process for everyone else. They run a very real risk of missing the White House deadline for compliance as a result.

By automatically generating each artefact from source code via a cloud-scale vendoring system, the ActiveState platform provides transparent provenance and custody chain, certifications for each package, and an SBOM that lists every piece of software. It helps manage and resolve complex open source dependencies automatically.

By eliminating the time, complexity, and danger associated with adopting open source, software developers may safeguard their supply chain and satisfy even the most demanding security standards by using ActiveState as a trusted vendor.

The S3 buckets may also store 'secrets', which include authentication keys, access tokens, and API keys, in addition to application data. These files may be source code or configuration files. Threat actors may be given much wider access to other services or even the company's corporate network if these secrets are unlawfully disclosed and made available to them.



Security researcher Eilon Harel learned that there are no tools for detecting unintentional data breaches during an exercise looking at SEGA's recent asset exposure. As a result, he decided to build his own automated scanner and publish it as an open source application on GitHub. Harel developed the Python utility called 'S3crets Scanner' to aid in the quick identification of exposed secrets on public S3 buckets. It automatically completes the following tasks:

- Gets a list of public buckets using CSPM
- Uses API requests to display the bucket's content
- Looks for any exposed text files
- Downloads the necessary text documents
- Looks for secrets in the content
- Sends outcomes to SIEM

Prior to downloading the text files for the 'secrets scanning' stage, the list is filtered to remove any buckets that were created with the intention of being made public. The Trufflehog3 tool, an enhanced Go-based variation of the secrets scanner that can check for credentials and private keys on GitHub, GitLab, file systems, and S3 buckets, will be used by the script while scanning a bucket to look at the text files' contents.

Siren and ShadowDragon collaborate to enhance investigation solutions

A strategic partnership between Siren, a provider of investigative intelligence analytics, and ShadowDragon, a supplier of open source intelligence (OSINT) unique data sets and APIs, has been announced. This partnership will make it easier for modern investigations to find, gather, and sort open source data.

The Siren platform is used in search and discovery operations as well as investigation solutions all around the world. It conducts enterprise-wide search and offers sets of connections that make it possible to clearly visualise and explore relationships between records. The range of data types utilised in open source intelligence is supported by Siren.

Siren may now resell ShadowDragon data, making it possible to obtain the data through straightforward contracts. Both on-premises and cloud implementations of the offering are supported. Numerous customers in the United States, Europe, the

Middle East, Asia, and Latin America employ ShadowDragon solutions. Fortune 50 businesses, as well as domestic and foreign law enforcement, government, military, and intelligence organisations, are included in this.

Analysts are helped by Siren's user-friendly interface as they look for significant bad guys. Siren combines information from various sources in real-time as events take place. The ability of Siren to onboard data using its fluid schema makes it suitable for the depth of open source intelligence provided by ShadowDragon. It is crucial to combine the new data with the current data in a seamless environment. Siren gives analysts the ability to combine old-school and cutting-edge forms of intelligence into a unified view.

Directus receives US$ 7 million in funding

In order to create an open data platform, which instantly adds an API and attractive no-code apps to any SQL database, Directus, an open source software business focused on democratising data across the organisation, has raised US$ 7 million in Series A financing. True Ventures led the round with participation from seed investor Handshake Ventures.



Bose, Adobe, and TripAdvisor are just a few of the companies that use Directus to power and grow their internal data tools and digital experiences. The company provides an expandable, fully open source platform that enables both developers and business users to maximise the value of the data within their organisation. Directus says it helps save development time for data-driven initiatives by 40 to 60 per cent by enabling anybody within an organisation to explore, manage, and display all business data.

A new or existing SQL database may be installed on top of the 100 per cent JavaScript-based Directus platform, which instantly provides business customers with a no-code app and strong developer tools without the need for any migration or proprietary vendor lock-in. Engineers gain from automatic REST+GraphQL APIs and a developer toolkit that has integrated authentication, granular permissions, unique data workflows, automation and alerting, reporting and analytics, among other features.

Merkle Tree Proof of Reserves solution is now open source

The Merkle Tree Proof of Reserves solution from the cryptocurrency exchange Gate.io is now open source, said Gate.io's founder and CEO, Dr Lin Han. The exchange welcomes other exchanges to obtain consulting and advisory support. It believes that other platforms can adopt its open source solution to promote transparency and industry health.

In 2020, Gate.io was the first to do Proof of Reserves audits in this way in collaboration with the US company Armanino LLP. The two recently finished another audit. According to a report by Armanino LLP, Gate.io has excess reserves of BTC and ETH, with respective holdings of 108 per cent and 104 per cent.

Since releasing the new audit this year, Gate.io has made a point of emphasising its dedication to boosting market confidence and has expressed the hope that other exchanges will do the same. The audit solution from Gate.io integrates cryptographic Merkle Trees, one of the fundamental building blocks of blockchain technology, with third-party certifications.

It promotes the openness of users' balances on exchanges by enabling them to openly and independently confirm their assets are included in the audit. It is now open source and freely usable by any exchange.



Netcracker receives Frost & Sullivan's Technology Innovation Leader award

The 2022 Company of the Year award for Asia-Pacific OSS/BSS and the 2022 Technology Innovation Leader award for Global OSS/BSS have both been given to Netcracker Technology. These awards honour Netcracker's leadership and top-tier technological innovation.

The analytics company listed a number of factors for the recognition, including fostering innovation in fields like OSS, RAN, private 5G networks, and multi-cloud. The numerous clients that Netcracker's BSS and OSS solutions have had a demonstrably positive impact on around the world were also mentioned by Frost & Sullivan, including Virgin Media O2, Zain Saudi Arabia, Vodafone Oman, Telenet, Globe Telecom, T-Mobile, Etisalat, Deutsche Telekom, and Sky New Zealand.

"Netcracker's 5G, cloud, and digital OSS/BSS service offerings have collectively helped CSPs drive the adoption of more innovative technologies, such as for the digital economy, by equipping them with the speed, flexibility and expertise needed to capitalise on rapidly evolving and emerging technologies, including 5G, Internet of Things (IoT) and industry automation," said Mei Lee Quah, associate director, ICT Practice at Frost & Sullivan.

"Receiving these awards from a well-respected analyst firm like Frost & Sullivan means a great deal to us," said Ari Banerjee, senior vice president of strategy at Netcracker. "We are working with many customers around the world, including the Asia-Pacific region – on large-scale digital transformation programs, BSS/OSS consolidation and creation and support of new business models – so these awards are a truly meaningful recognition of our efforts."

The money comes at a time when the company is experiencing enormous, worldwide growth. Last year, the Directus team expanded from two to over twenty, with hundreds of contributions.

Open source is flourishing: GitHub report

Some unexpected new trends that are influencing and reshaping the developer community have been outlined in GitHub's most recent report on the state of open source software. The survey concluded that Australia is one of the fastest-growing developer communities in the Asia-Pacific region, and that many other communities all over the world are also growing in size, despite the fact that the developer environment still faces a number of obstacles.



According to the report's statistics, 94 million people use GitHub, and 90 per cent of businesses employ open source software. Additionally, there were 413 million contributions to open source in total in 2022. Infrastructure as Code (IaC), the growth of open source communities, and OSS contributors and their preferences were the three key trends in the research.

HCL was found to be the fastest-growing language, with developers also using Shell and Go language in 2022. There was an uptake in commercially backed projects, and successful open source projects saw salaried developers make regular contributions on behalf of their employers.

The research indicated that new OSS contributors tended to favour projects with financial support from businesses. The figures showed that 50 per cent of first-time contributors worked on commercially sponsored projects.

Speedb open sources its data engine

Speedb (*www.speedb.io*), which created the RocksDB drop-in replacement Speedb data engine, is making the technology it has been providing in its enterprise edition for the previous two years available as open source.

With Speedb's open source data engine, the developer community has a chance to influence the data architecture of the future. Users and collaborators of Speedb and RocksDB can interact and exchange information, problems, and solutions in the Speedb open source community.

RocksDB, a derivative of Google's LevelDB that Facebook released as open source in 2012, is the *de facto* industry standard for embedded key value stores (KVS). The Speedb community fills a void in the KVS arena as RocksDB popularity grows to include a wider range of use cases and deployment circumstances by offering developers new ideas, support, and services to meet changing needs.

RocksDB is widely used, but its scaling issues necessitate sharding and performance tuning, which take up excessive amounts of developer time as applications expand. The Write Amplification Factor (WAF) of RocksDB was reduced from 30 to 5 thanks to a compaction method that Speedb developed, along with the database's I/O and flow algorithms.

Speedb will collaborate with the open source community to tackle problems like the expansion of metadata, which is having an increasingly negative influence on the performance of databases, streaming apps, AI/ML, cybersecurity, analytics, and more. Users of RocksDB can incorporate the Speedb data engine into their apps with a few lines of simple code changes to improve metadata memory performance and free up time-consuming operational duties.

Netflix moves to GPAC open source software from Motion Spell

After an 18-month transition period, Netflix is implementing GPAC open source software from Motion Spell across its global content operations. Open source software called the GPAC Project on Advanced Content offers a modular multimedia framework for packing, broadcasting, examining, and playing content. It offers a variety of tools for handling media content and places a big emphasis on standards.



Netflix's research and development team collaborated with Motion Spell, a media technology software and consulting firm, for the implementation. The streaming platform needed a reliable packaging solution that was versatile and future-proof. The GPAC open source software is licensed exclusively to Motion Spell for commercial use.

Romain Bouqueau, CEO of Motion Spell, said, "Netflix is a company with an exceptional team, clear goals and an innovative R&D programme. Combining our expertise made the transition process to GPAC very efficient. We're delighted that Netflix has gained in flexibility, while GPAC has gained a customer who respects the value of open source software."

Anne Aaron, director of encoding technologies at Netflix, added, "As part of our strategy to build a flexible and efficient packaging platform, we needed a strong partner with deep expertise of the best encoding and packaging tools. Working with Motion Spell helped make the entire process of integration incredibly easy."

For more news, visit
www.opensourceforu.com

Swisscom and Netcracker extend collaboration

Telecom provider Swisscom has expanded its engagement with Netcracker for enhanced professional services and OSS (operations support system) to support converged B2C and B2B services.



In order to serve the operator's new IP transport domain, Swisscom will develop its use of Netcracker Digital OSS, including Active Resource Inventory as well as Service Management and Orchestration, and Hybrid Resource Management. To support the new engagement, Swisscom is also utilising Agile product delivery and Netcracker's product support. Swisscom will gain from reduced OpEx and a speedier time to market by focusing on self-sufficient DevOps and configuration capabilities for the IP transport sector.

"We've experienced a lot of success over the years through our strong partnership with Netcracker," said Peter van der Lee, head of IT at Swisscom. "With this further commitment, we are looking forward to supporting new lines of business and our journey to the cloud."

"Netcracker is truly humbled to have the trust of Swisscom through many different engagements," said Benedetto Spaziani, GM at Netcracker. "As we expand our offerings with Swisscom and facilitate the path to the cloud, we are grateful for the opportunity to work with a modern and innovative company."

State of Open Source: Trends, Insights and Lessons

The open source world is in a state of constant evolution, with new trends emerging continuously at a pretty quick pace. Here's a summary of what has been happening in the recent past in this world, and especially at GitHub, to help developers analyse what the future holds.



GitHub, as we all know, is a code hosting platform where the world builds software. Over 83 million developers and more than four million organisations are using it today. The open source ecosystem itself is seeing more than 2.6 billion contributions per year, thousands of open source communities, and more than 200 million private and public repositories. So, essentially, this is where a lot of the activity around open source development is happening today. Just last year, over 16 million new users joined the platform. About 61 million new repositories and 170 million pull requests also got merged.

India is witnessing a lot of open source contributions as well as development in this space. There are more than 8 million users of GitHub in the country. Over 200 million contributions to open source came from India in the last year alone. It is also important to note that more than 10 million software projects around the world depend on packages that were created in India.

Let us now talk about communities, documentation, and about developer experience along with the latest trends.

Communities

GitHub welcomed more than 1.4 million new contributors to the open source community last year. This has allowed it to continue to build software and pass it across the world. In 2021, there were more than three million people who were contributing to open source projects for the first time, and this number only seems to be growing. Good mentorship can really help these first time contributors and is an important community asset. When new contributors get friendly and timely reviews, and there is a commitment to mentorship from the team, there is a positive effect on productivity.

A healthy, collaborative and strong culture can actually reduce the need for mentorship itself. People who are interested in open source can get involved through special interest groups, not just deep technical communities. If you look (refer to Figure 1) at the percentage of contributors who have opened an account over the past two years, you can see a lot of repositories that are getting developers started with their open source journey. The topics in these repositories include gaming, manga, science education, and much more. Developers tend to first contribute in these areas and then move on to the deep technical community.

Documentation

Documentation is key to getting any developer or open source contributor started. Regardless of the format, documentation is important. Building on top of someone else's work in a community can help accelerate one's growth. Documentation also often

Repositories	% contributors with account age <2 years
gustavoguanabara/git-github	72.9%
trustwallet/assets	64.1%
flybywiresim/a32nx	58.9%
progedu/pull-request-training	55.2%
progedu/pull-request-study	54.6%
tachiyomiorg/techiyomi-extensions	47.1%
google/it-cast-automation-practice	47%
learn-co-students/first-ide-lab-js-into-000	44.2%
learn-co-students/javascript-logging-lab-js-into-000	44.2%
Chia-Network-chia-blockchain	44.6%
techiyomiorg/techlyomi	36.3%
TheAlgorithms/Python	36.6%
files-community/Files	35.8%
backend-br/vagas	33.5%
MicrosoftDocs/mslearn-tailspin-spacegame-web	32.6%
mercadolibre/api-rules	29.7%
ppy/osu	29.4%
MicrosoftDocs/microsoft-365-docs	29.4%
qgis/QGIS	29.1%
mdn/browser-compat-data	29.8%

Figure 1: New commerce initial attraction to open source community

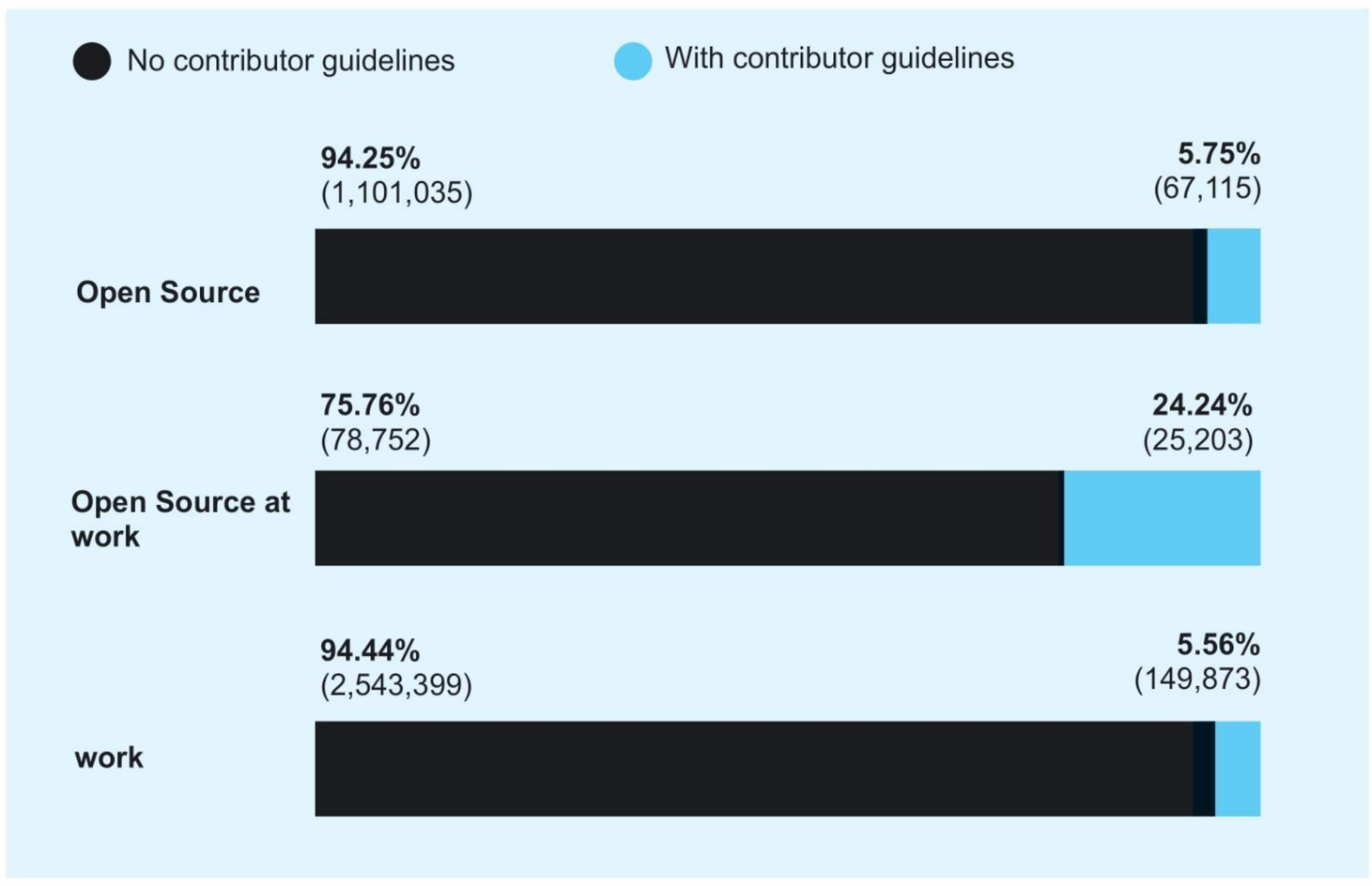


Figure 2: Repos with and without contributor guidelines

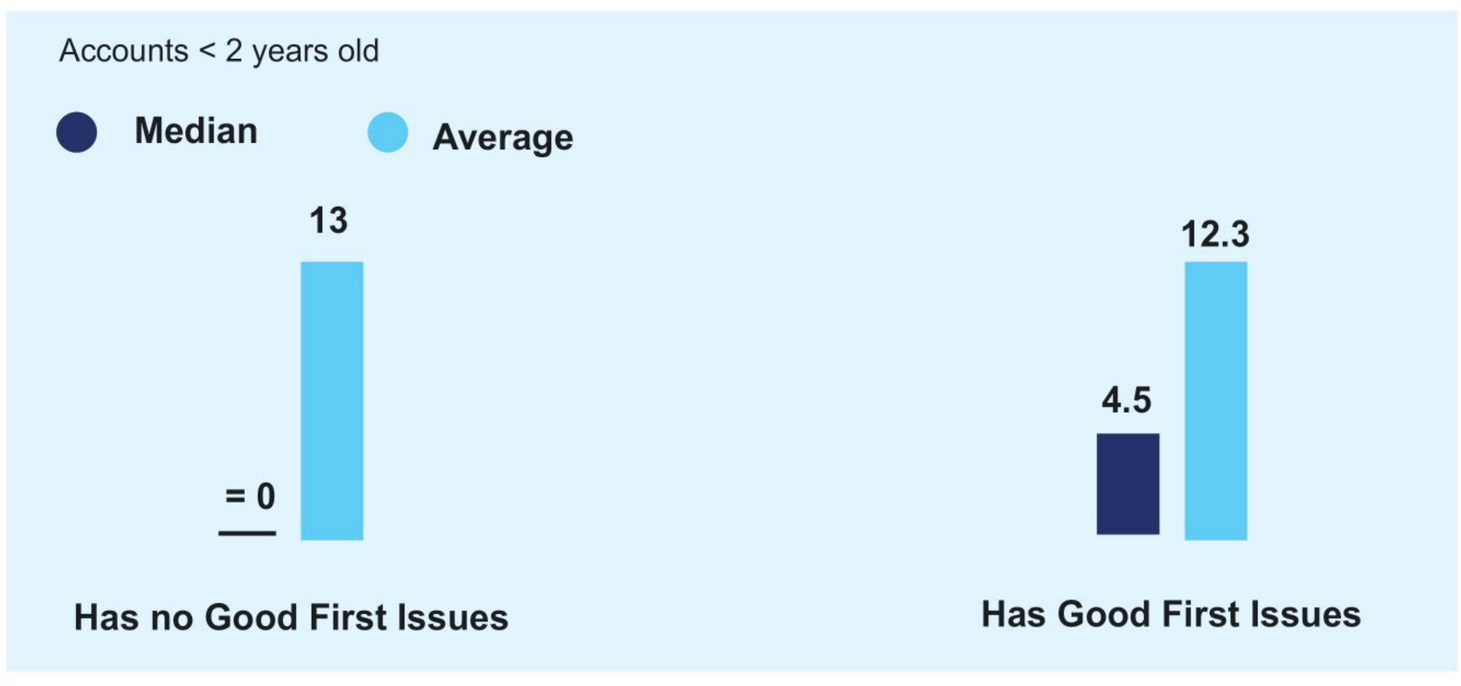


Figure 3: Percentage of new contributors depend on whether a repository has any Good First Issues

indicates if a repository is reliable to reuse code from, and whether the project is still active or not. A developer considers documentation trustworthy when it is up-to-date, and has a higher number of updates from others in the community.

Open source projects use README, contribution guidelines, GitHub issues and a lot of other things to elevate their quality, and also to share information that makes them more attractive to new contributors as well as enterprises. Developers see up to 50 per cent boost in productivity when the documentation is up-to-date, especially when it is detailed, reliable, and comes in different formats such as articles, videos or forums.

Documentation is often under-invested despite being so critical. Internal documentation could be associated with onboarding, governance, tutorials, best coding practices, and a whole lot of other things. It also creates the rules of engagement, which remove ambiguity and friction. When companies work closely with the open source community, it facilitates building on each other's work. Without this, there can be a whole lot of confusion as to how a certain thing is supposed to be done (refer Figure 2).

Median hours to merge pull request

Let's look at the median hours to merge pull requests where the pull request had at least one reviewer (Figure 4). There is better growth in open source where the median hours are around 11 hours. But you can see open source at work growing all the way up to 10 hours. With work, it is coming back to the pre-pandemic levels.

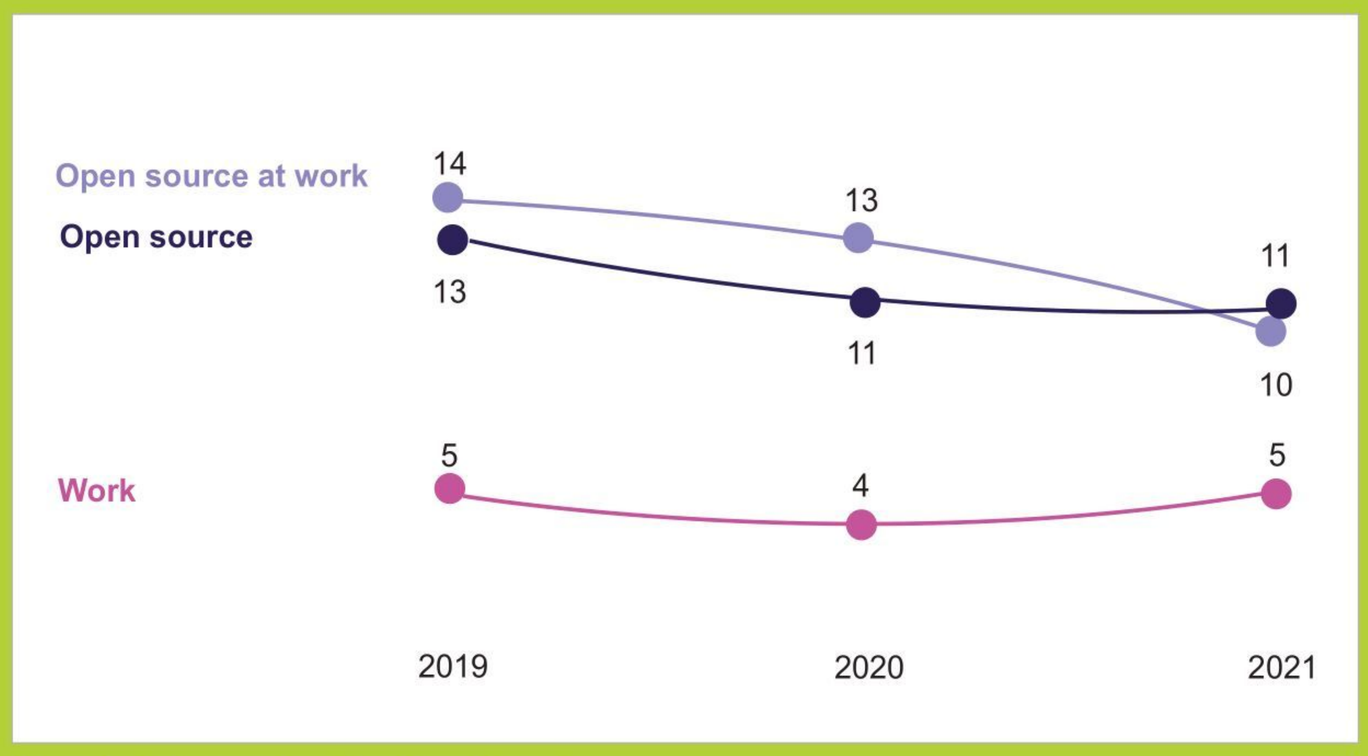


Figure 4: Pull requests with at least 1 reviewer

The stats in Figure 2 show the percentage of repositories with and without contributor guidelines. You may be wondering as to why a company should add contribution guidelines to its repositories. This is because these guidelines remove ambiguity and friction, and give a 17 per cent boost to productivity when developers know how they can contribute to repositories that belong to a different team or a community.

Open source repositories often use READMEs to share information and also to signal that the community is active. It is important to note that work repositories are used considerably less than open source projects, which may be because they share documentation and information in other internal forums.

GitHub issues are documentation too. Creating issues is the most common contribution. It invites non-coder contributors as well. This provides a great way for people to join, get help as well as manage projects. People can create issues within the first hour of joining GitHub. There is a way on GitHub itself to help you get started with this; it is called Good First Issues and is a great way to guide new members with their first contribution. Large repositories are most likely to use this. Repositories where almost 21 to 30 per cent of issues are not good generally have 13 per cent new contributors. You can see in Figure 3 the percentage of new contributors when a repository has Good First Issues and when it has no Good First Issues. This is 12 to 13 per cent more on average when you mark an issue as a Good First Issue.

Fast is better

Every developer, manager, and organisation wants tools and processes to be fast and easy. Similarly, open source project leaders and maintainers look for ways to make communities more welcoming. One of the ways to increase productivity is to improve the code and automate it, as it increases the overall speed of the work. It also improves collaboration and work culture. Good automation helps teams to communicate better and clearly, something that has been proven by many studies from time to time.

Developers' patterns on GitHub reflect that automating software delivery is a key enabler in open source and also helps teams to scale faster. Large repositories use features like GitHub Actions far more than small- or medium-sized repositories. When large repositories start using GitHub Actions, teams tend to merge two times more pull requests per day than before, which is almost a 61 per cent increase, and they merge 31 per cent faster. Repositories that are using GitHub Actions increase the number of merged pull requests. This increases the number of merged pull requests by 36 per cent while shrinking the merge time by 33 per cent.

You can improve your ability to merge pull requests quickly in the following ways:

- Assign no more than three reviewers to a pool request in an open source repository. This increases the chances that it will get merged within 24 hours.
- At work, pull requests with just one review are often merged within an 8-hour workday.
- Automation and community help us build better, together.

I hope these quick trends will help you understand what is really going on in the open source industry and where the contributors are coming from today. It is important to understand the relevance of documentation, work culture and the overall developer experience to repurpose your workflow accordingly. END

By: M.V. Karan

The author is senior manager, International Developers Relations, GitHub. This article is put together from a tech talk he gave at 'Open Source Summit: Frontiers of Innovation 2022', an event co-hosted by Samsung and IEEE Bangalore.

Transcribed and curated by Laveesh Kocher

EFY GROUP
YOURS SINCE 1969

Wanna Support Open Source For You?

Subscribe to the Magazine— so that we can keep promoting Open Source.



AMAZING OFFER

Pay for 6 Issues Get 12!

(Buy One Get One Free!)

WORLD'S LEADING PUBLICATION ON OPEN SOURCE

TEAR OUT — **ORDER FORM** — TEAR OUT

Please Tick (✓) Your Choice

☐ **Pay for 12 Issues Get 24 Issues**
and save 50%
(Buy 12 Issues Get 12 Issues Free!)
₹1440

☐ **Pay for 24 Issues Get 48 Issues**
and save 50%
(Buy 24 Issues Get 24 Issue Free!)
₹2880

☐ **Pay for 36 Issues Get 72 Issues**
and save 50%
(Buy 36 Issues Get 36 Issue Free!)
₹4320

To subscribe online, visit
https://tinyurl.com/y5kuv4la

OR **SCAN THIS CODE**

Name_____ Organisation_____ Mailing Address_____

_____ City_____

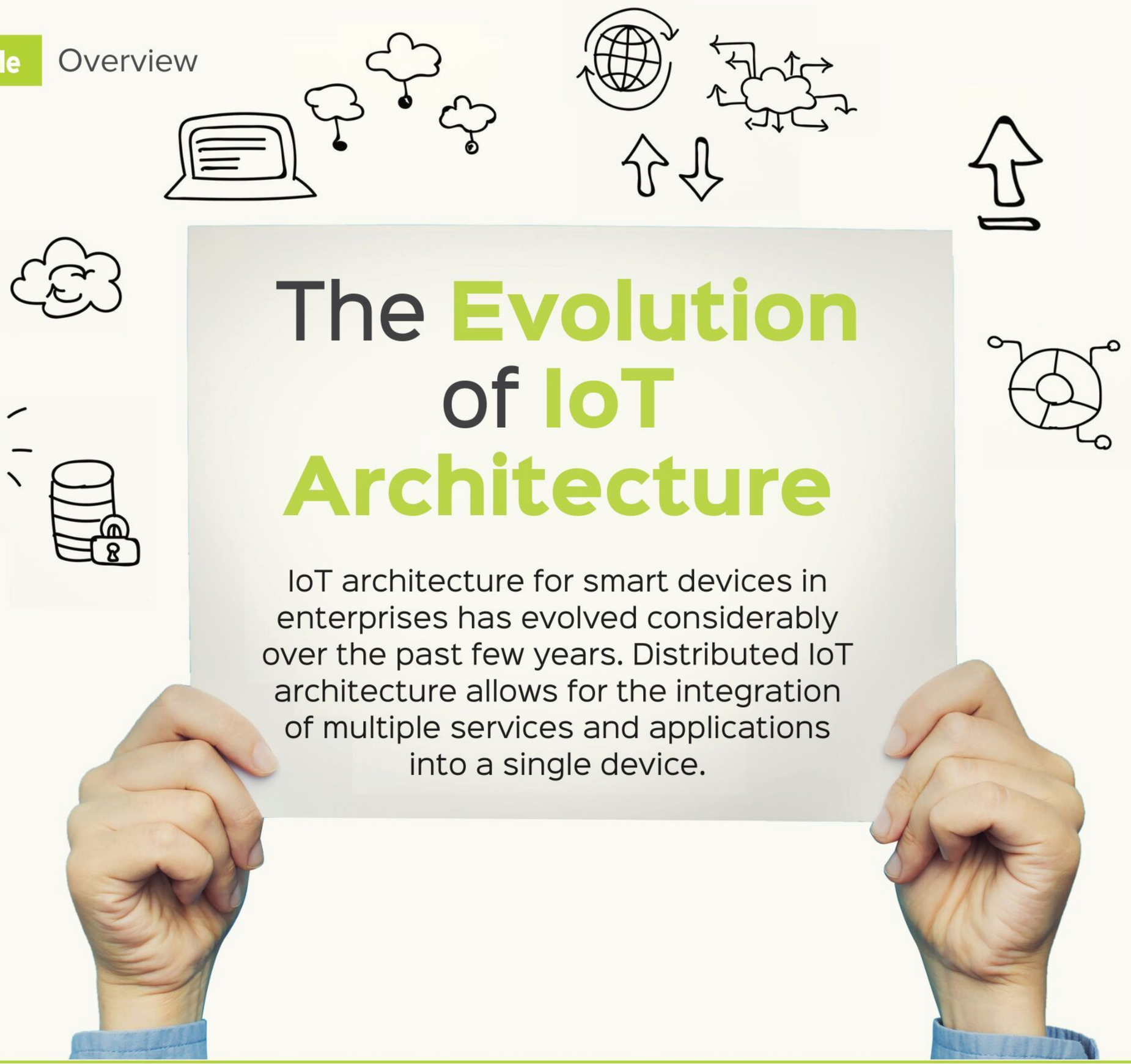
Pin Code_____ State_____ Phone No._____ Email_____

Subscription No. (for existing subscribers only)_____. **I would like to subscribe to the above (✓)marked Open Source For You magazine starting with the next issue.** Please find enclosed a sum of Rs_____by DD/MO/crossed cheque bearing the No._____ dt._____in favour of **EFY Enterprises Pvt Ltd,** payable at Delhi. *(Please add Rs 50 on non-metro cheque)*

Please mark one (nearest) relating to your subscription: ☐ **Indian Company** ☐ **MNC** ☐ **R&D organisation** ☐ **Engineering institute** ☐ **College/School** ☐ **Any other (specify):**_____

Send this filled-in form or its photocopy to : **EFY Enterprises Pvt Ltd,** D-87/1 Okhla Industrial Area, Phase 1, New Delhi 110 020 | Ph: **011-40596600** | e-mail: **support@efy.in**

Terms:- # These rates are applicable for new subscribers as well as renewal by existing subscribers. # Can access ezine till your subscription is active # The rates are valid for subscribers within India only. # Please allow 4-6 weeks for processing of your subscription. # The subscription copies will be dispatched through ordinary post only # Subscription Agents will not get agency commission against this scheme # Disputes, if any, are subject to exclusive jurisdiction of competent courts and forums in Delhi/New Delhi only. * Replacement will be made if intimation of damaged / non-receipt of copies is received within 30 days of its publication ** After three months, if you are not satisfied with the magazine, your balance amount will be returned (Not applicable for gift offer)



When I bought my first IoT-enabled light bulb and controlled it with my phone about five years ago, I thought it was the coolest thing and that I would cherish it for a long time. That thought changed pretty soon, though. Today I have over 300 IoT devices at home, which are controlled with numerous apps. I manage these by using a spreadsheet that maps each device to its app so I don't go crazy having to memorise all the device controls. I sincerely hope advancements in IoT architecture will enable simpler and integrated ways of managing devices.

IoT architecture comprises the numerous elements that make up an IoT networking system like sensors, protocols, actuators, cloud services, edge services, data services, and applications. The basic three-layered architecture is easy to implement. It helps to collect data, process it, and then transform it into usable information. Embedded analytics with machine learning and artificial intelligence enables businesses to remain agile. The three layers are:

- *Perception* for the sensors, gadgets, and other devices
- *Network* for the connectivity between devices and applications
- *Application* for the user interaction and responses

IoT architecture makes all this possible by ensuring that data is handled properly and gets to the right place. Networks would become unstable without adequate IoT architecture, undermining the very point of investing in IoT in the first place.

Although this 3-layer architecture identifies key layers and is the simplest to implement, it does leave out details that are critical in large-scale enterprise-grade IoT solutions, especially security threat concerns needing specific attention across each of the layers. This has led to the evolution of the IoT architecture into 4, 5, 6, and 7-layer architectures.

Four-layer architecture is like the basic 3-layer architecture with one key additional layer breaking down the network connectivity layer and creating a new support layer. Adding the new support layer between the connectivity layer and the perception layer improves the ability to manage data efficiently and its security.

The 5-layer architecture is one of the most adopted and implemented architectures, because of its simplicity. It gives the right amount of balance, allowing greater management complexity along with the ability to scale the number of IoT devices in use. Here, the collaboration layer is separated from the application layer. This makes IoT devices much more lightweight, as the processing of data is moved into the application layer that could be deployed on the cloud or edge computing.

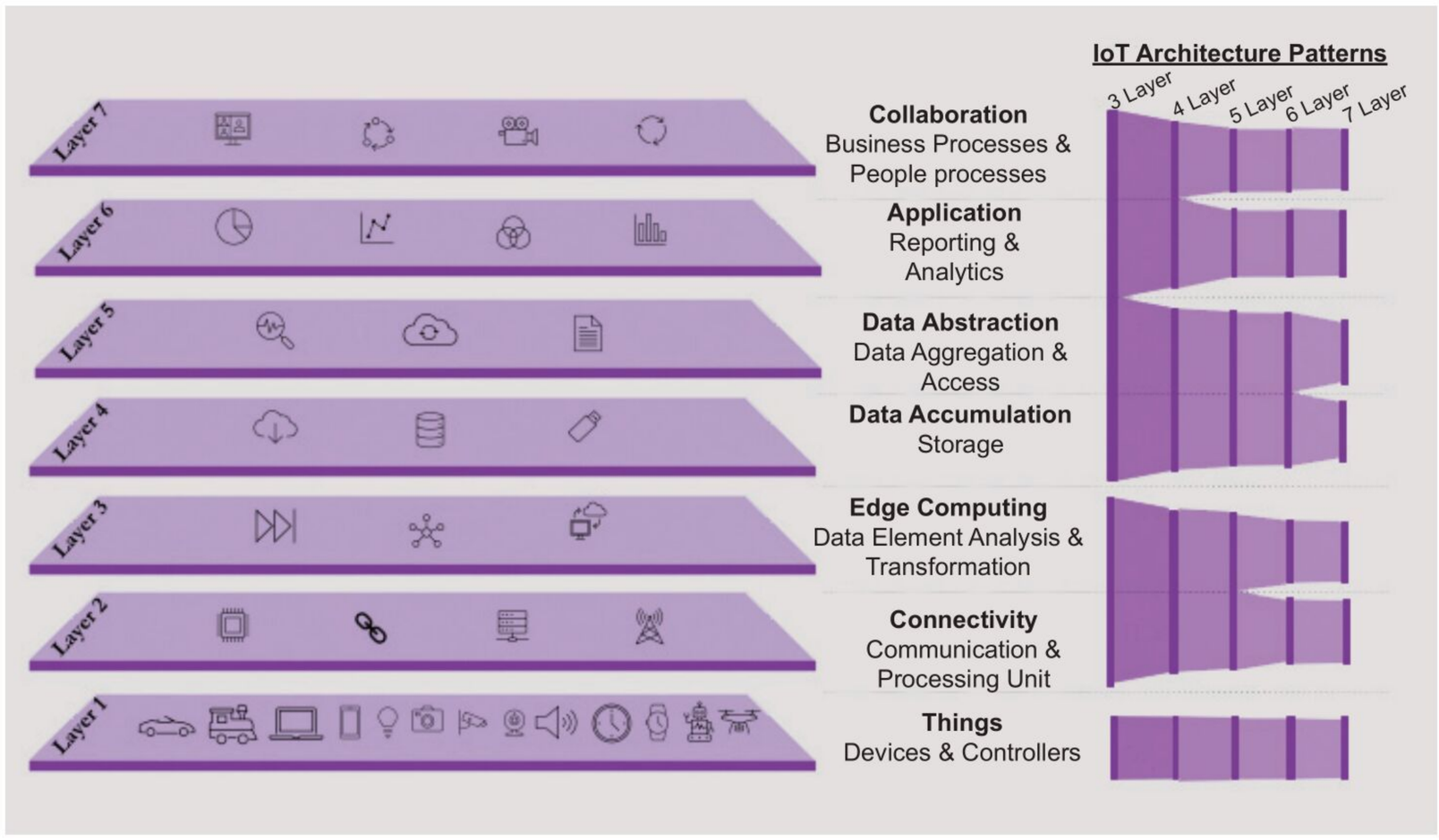


Figure 1: IoT architecture reference model and layered patterns evolution

The 6-layer architecture breaks down the network connectivity layer, creating an edge layer that enables efficient transport of data closer to the device. This improves the performance and utilisation of the device, and increases its speed and accuracy, specifically in transportation and logistics types of use cases.

Lastly, the most advanced IoT architecture model at present is the 7-layer architecture, which separates data accumulation and data abstraction. With this separation, the cloud plays a key role in the ever-increasing data collection journey of IoT devices, especially in the case of enterprises that deploy a massive number of devices for their customers and consumers. In data abstraction, layer data aggregation and access management for the application are primary concerns. In the data accumulation layer, the storage of data is the primary focus as it helps classify the quality and quantity of data stored by usage.

The evolution of IoT architecture along with commercial and open source solutions has made possible what was unimaginable just a few years ago.

Distributed services architecture

Distributed services architecture (DSA) enables smart communication, logic, and applications at all layers of the IoT infrastructure. The intent is to combine the various devices, services, and applications into a structured and flexible real-time data model. Open source DSA aims to create a network of manufacturers, creators, and solution providers who will add to a growing collection of distributed services that enable protocol translation and data integration to and from external data sources.

DSA is designed to connect upstream rather than downstream, making it easy to connect distributed services on a private and a public network. Decentralised services and device-oriented interactions are made possible by DSA. A network architect can divide functionality across separate computing resources using this architecture. The system can be scalable, fault-tolerant, and utilise all computing resources available to it from the edge, the data centre, the cloud, and everything in between thanks to a network topology

Advancements in enterprise 5G networks have increased the reliability of IoT architecture and these networks now process data faster than ever before. 5G addresses the implementation difficulties IoT architectures faced with spotty cellular networks, unreliable bandwidth, speed limits or unknown cost forecasting. Although designs for IoT architecture are maturing faster than connectivity speed, the improved speed has given use cases across healthcare, agriculture and manufacturing a renewed purpose.

The rapid adoption of IoT devices is just one dimension of how hungry consumers of the technology are for smarter devices, creating a significant uptick in the development and deployment of IoT device-related technologies. Let's hope that in the coming days there is a rapid unification of standards for leveraging the distributed services architecture of IoT, making its development and adoption smoother for everyone involved. END

References

- The Open Source Plattform & 'Toolkit' for Internet Of Things Devices, Services and Applications, *https://iot-dsa.org/*
- IOT One Distributed Services Architecture, *https://www.iotone.com/supplier/distributed-services-architecture-dsa/v1594*

By: Bala Kalavala

The author is a seasoned emerging technical architect, evangelist, thought leader and a sought-after keynote speaker. He currently works as a distinguished member of the technical staff and heads the Enterprise Architecture practice as chief architect in a global technology consulting firm.

Innovative Lessons from Ladakh for Rest of the World

While the world wants to indigenise high-tech, Ladakh uses simple, local technologies to solve most of the problems. From solar-powered houses to mountain-to-mountain connectivity, Ladakh shows how simple methods can achieve huge milestones. This is a story not just about sustainable but regenerative development.

Living in Ladakh might seem like living in a remote part of the planet, especially when people feel cut-off from rest of the world for up to seven months in a year. With roads blocked due to heavy snow, landlocks exist in Ladakh till May or June every year, showing how remote the place is.

While problems vary from place to place and terrain to terrain, everyone is looking for simple and local solutions here to indigenise and solve them. Sonam Wangchuk from Ladakh believes, it would be interesting to know how the same science is being applied differently in Ladakh. He further explains how new digital technologies can be used to connect mountains, marking a huge milestone in network and connectivity.

Ladakh, located on the northernmost part of India, across the Himalayas, is quite a high and dry place with temperatures ranging from -35°C to +35°C from winter to summer. Situated on the Tibetan plateau—the rooftop of the world—this place is as close as it can get to outer space while still being on planet Earth. In terms of climate, Ladakh is considered a rarity, since when water freezes, problems like freezing, blockages, and bursting open of pipes happen. Yet, Ladakh is a thriving hub of innovation and problem solving.

> "We rescue it (the straw) and save it from polluting Punjab and Delhi. They become insulated blocks in Ladakh that are used to build the solar powered houses, thus cutting pollution in both Punjab and Ladakh."
>
> ***—Sonam Wangchuk***

Solar-powered houses

The launch of Students' Educational and Cultural Movement of Ladakh (SECMOL) has enabled children from the mountain range to have hands-on experience and learn things the practical way. Built entirely with mud from under the feet and powered by the sun over the head, this is one of the first schools that has had a zero energy off-grid campus.

With abundant sunshine available almost up to 300 days a year, solar-heated mud houses have been built in Ladakh. Despite their low cost of construction, these houses are efficient enough to provide +18°C environment in a -20°C winter, with nothing but the sun to help.

Himalayan Institute of Alternatives, Ladakh is also working on some local solutions. "One of the interesting experiments is where we've built



Figure 1: The solar-powered and solar-heated SECMOL

EFY GROUP
YOURS SINCE 1969

Stay **Connected.**
Stay **Informed.**
Stay **Ahead.**



SUBSCRIBE AND **SAVE**

ORDER FORM

PRINT MAGAZINE	1 YEAR (12 copies each)	3 YEARS (36 copies each)	5 YEARS (60 copies each)
		WITHIN INDIA (IN RUPEES)	
Electronics For You (Rs 100/copy)	840 ☐	2150 ☐	3000 ☐
		SAARC COUNTRIES (IN US$ BY AIR)	
	50 ☐	135 ☐	NA
		OTHER COUNTRIES (IN US$ BY AIR)	
	100 ☐	270 ☐	NA

e-magazine subscriptions within India are available at half the rates mentioned here.
Overseas rates for each e-magazine in US$: 1 year: $12; 3 years: $33; 5 years: $50 only

PRINT MAGAZINE SUBSCRIBERS GET:
- Free e-magazine every month
- Free delivery of print magazine by post
- And much more (check: subscribe@efy.in)
- For delivery by courier, please add Rs 50 for each copy

To subscribe online, visit https://payment.efyindia.com | OR | **SCAN THIS CODE**

Name_____Organisation_____Mailing Address_____

_____City_____

Pin Code_____State_____Phone No._____Email_____

Subscription No. (for existing subscribers only)_____. **I would like to subscribe to the above (✓)marked magazine(s) starting with the next issue.** Please find enclosed a sum of Rs_____by DD/MO/crossed cheque bearing the No._____dt._____in favour of EFY Enterprises Pvt Ltd, payable at Delhi.

Please mark one (nearest) relating to your subscription: ☐ **Indian Company** ☐ **MNC** ☐ **R&D organisation** ☐ **Engineering institute** ☐ **College/School** ☐ **Any other (specify):**_____

Send this filled-in form or copy to : EFY Enterprises Pvt Ltd, D-87/1 Okhla Industrial Area, Phase 1, New Delhi 110 020 | Ph: 011-40596600 | e-mail: support@efy.in

Terms:- # These rates are applicable for new subscribers as well as renewal by existing subscribers # Please allow 4-6 weeks for processing of your subscription.
Please include your pincode for prompt delivery of your copy.

four passive solar-heated (PSH) mud buildings and obtained fascinating results," according to them.

The graphs in Figure 2 compare the difference in temperature for a PSH house, a fossil fuel heated building, and those buildings that are not heated. They are represented by the orange, maroon, and bluish-grey lines, respectively. With the month of January being the coldest, the average temperature inside the solar-powered houses ranges from +18 to +25°C, even though the temperature outside is -15°C.

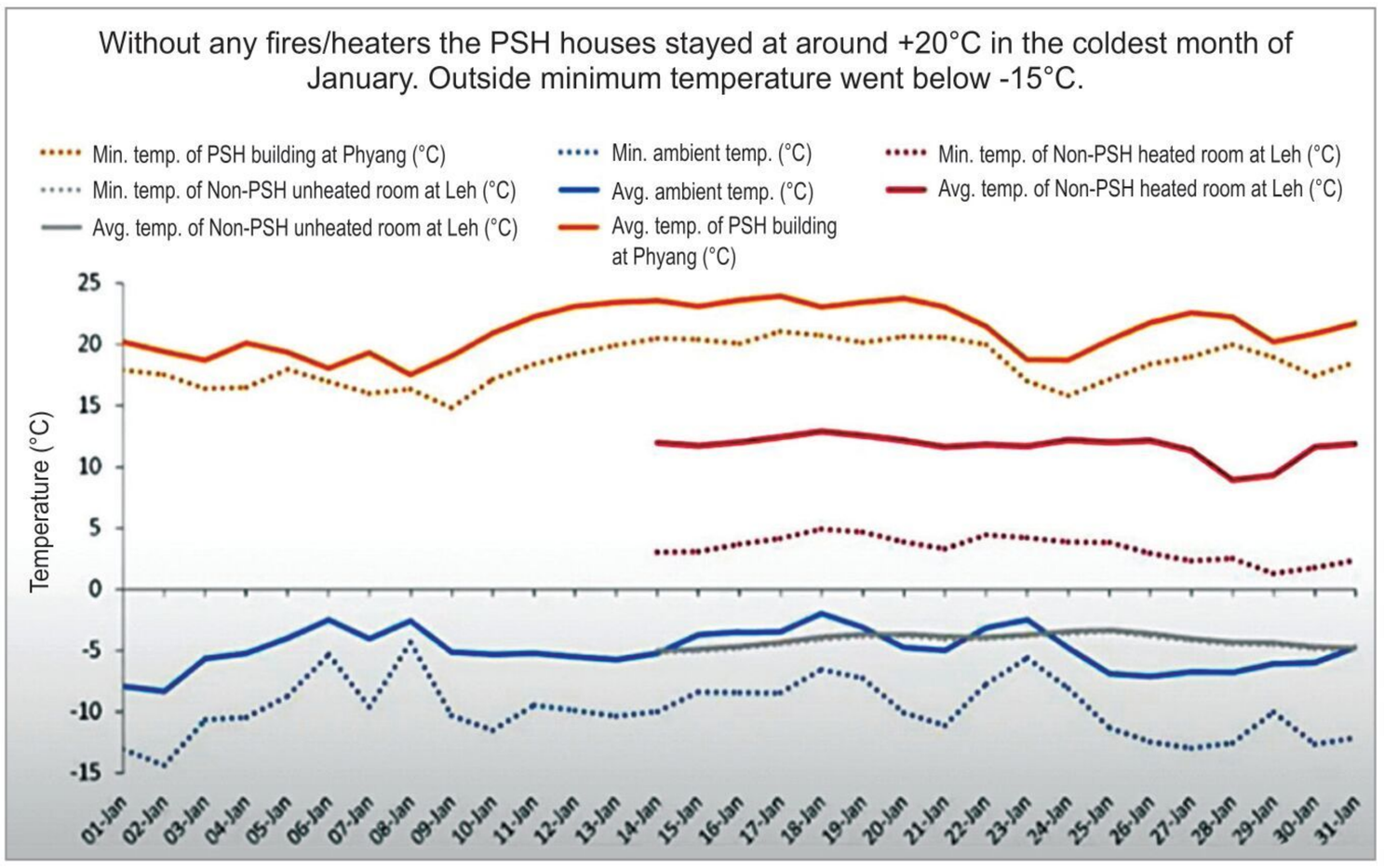


Figure 2: How PSH houses are more effective than the others

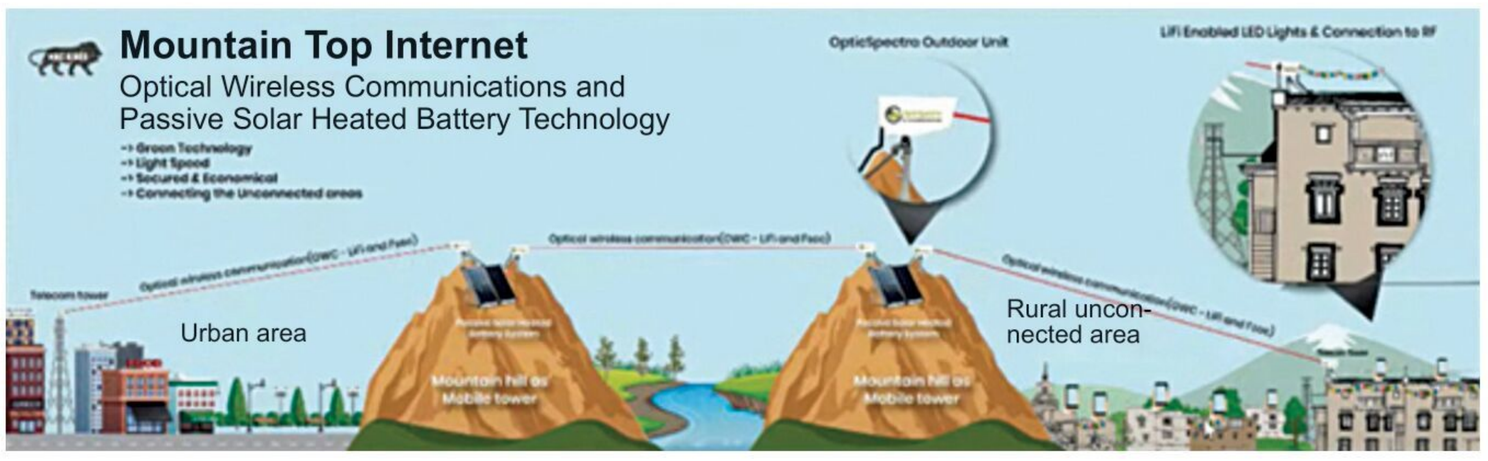


Figure 3: Working of the mountain-top internet using line-of-sight propagation

Construction

The PSH buildings are built using straw and clay. The latter is quite abundant in Ladakh, while the former comes from the state of Punjab, where it is seen as a source of pollution when burnt. Sonam says, "We rescue it (straw) and save it from polluting Punjab and Delhi. When the clay and straw are mixed, they become lightweight insulated blocks that are used in Ladakh to build the solar-powered houses, thus cutting pollution in both Punjab and Ladakh."

The buildings are generally oriented towards the south where the winter sun roams most of the time. Behind the glazing of the southern wall, a dark wall is erected for absorbing the light waves (and the heat), thus capturing the solar energy, and storing it in a sort of no-cost battery that is made of plastic waste bottles. The water in the plastic bottles traps the heat during daytime and releases this heat at night.

To retain the precious heat indoors, a good amount of insulation is required in the three other side walls as well as the roof and floor. This insulating material is made from the wood and wool waste obtained from the Pashmina industry. Their latest experiment is using solar-powered mud principles along with RCC frame structures to obtain the conventional benefits along with solar passive heating.

The amount of oil burned on the Indian borders of China and Pakistan is immense. It is used for heating the army shelters at around 3500 to 4575 metres (11,500 to 15,000 feet) altitude. However, it produces 330,000 tonnes of carbon-dioxide. These are now going solar, with the shelters being redesigned to maintain the room temperature between +12 and +15°C, when the outside world is at -22 to -30°C.

Ice Stupa artificial glacier

Sonam says, "With global warming melting away glaciers, people laughed when we spoke of artificial glaciers." The principle is to freeze the stream of winter water that goes unused as there is no farming in that area. A pipe is placed upstream and brought down to where the fields are. Since water tries to flow downwards through the pipe, the water gushes out at lower end of the pipe without any power used. Because of the -20°C or so temperature outside, the water starts freezing in the form of a cone below the pipe, thus forming an Ice Stupa.

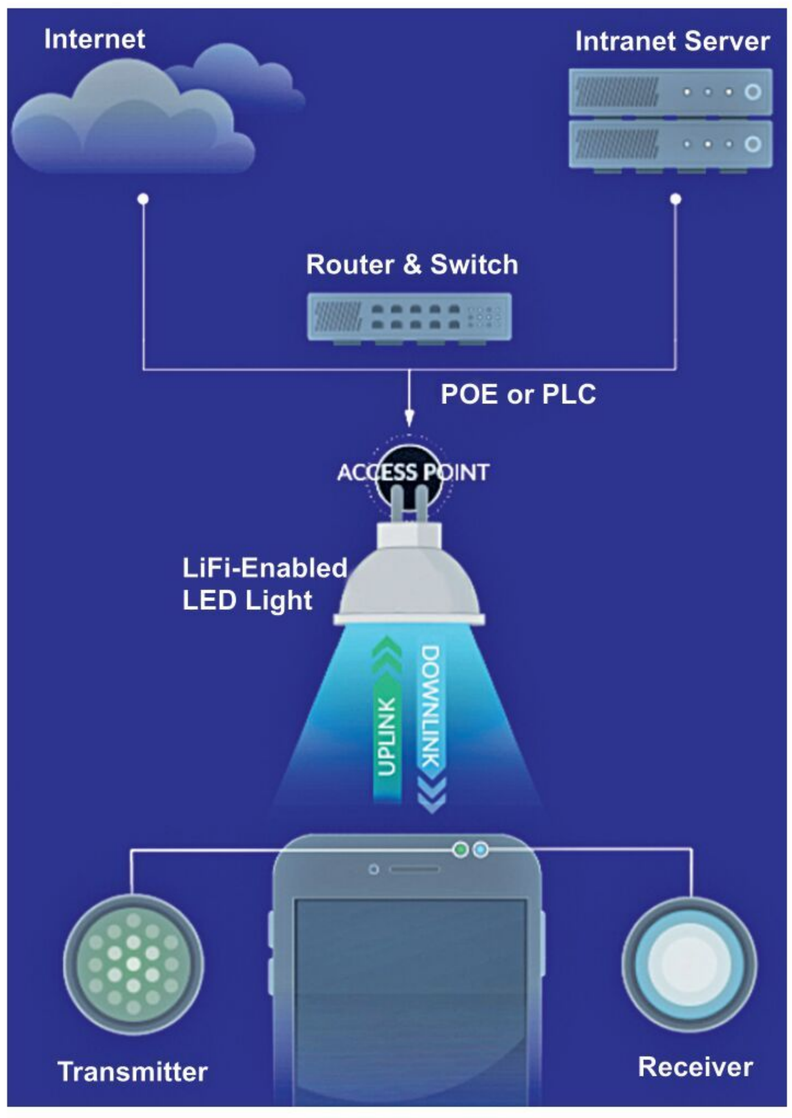
Since the Ice Stupa is in a conical shape, it can store a great amount of water, offering a very less surface area for evaporation. As the Ice Stupa slowly melts, the water is stored in a tank and is used for farming using the drip irrigation method.

Sonam adds, "It is simple with no high-technology power. It has become a movement in villages with competitions being held nowadays for the largest and biggest Ice Stupas. This innovation is expanding its borders and spilling outside India into Nepal, Pakistan, Chile, Switzerland, Czech Republic, Europe, and so on—both for aesthetics and winters as well."

Li-Fi—the world's fastest Wi-Fi

Li-Fi is a wireless internet communication system that uses visible light for transmission at very high speeds. Combining the world of electronics and photonics, this technology captures data in modulated light frequencies of an LED light source and transmits them. The photosensitive detector converts the light frequency signal back to an electronic data stream, allowing faster bi-directional wireless communication.

Like decoding Morse code, this technology has a much faster rate of decoding information, which is up to millions of times per second. In terms of transmission speed, Li-Fi can go over 100Gbps, which is nearly 14 times faster than even WiGig.



The basics of a Li-Fi system
(Credit: https://Li-Fi.co/what-is-Li-Fi)

Using Li-Fi technology, Sonam Wangchuk conducted a live Youtube session on April 1, 2022 and displayed the speed of the internet.

In terms of geography, the signal from the Jio tower cannot be received nor seen at the SECMOL school. However, it can be seen and received on the SECMOL mountain (Li-Fi Tower1 hub), which is at an altitude of 800m. After the signal is captured at the Li-Fi Tower1 hub as a line-of-sight (LOS) propagation from the Jio tower, it transmits the signal to the optical communication unit (Figure 6) via light modulation (laser beams). This unit is placed on the roof of the SECMOL school and the signal from the Li-Fi Tower1 hub is decoded and converted back to an electrical signal. Hence the signal from the tower can reach remote areas (indirectly), as seen in Figure 6.

Li-Fi eliminates the cost for digging of optical fibre cables and building of 10-story tall towers. However, powering the towers can be a challenge considering the cold atmosphere on the mountain top. Major telecom providers say that the power required for each tower is up to 7kW. Optical units on the other hand require only 48W.

Just like the solar-heated building, Navtech, HIAL, and SECMOL have developed a solar-heated battery storage device. When the temperatures in the barrel (below the solar panel) were measured, a difference of approx. 61 degrees was obtained. Sonam says, "The temperatures in the night are a minimum of +26 degrees, even while it is a chilling -40 degrees outside. This proved the power system can work unmanned on mountain-tops."

As shown in Figure 4, using Li-Fi receivers and transmitters, any signal (example, telecom) can be obtained from a normal tower. When a laser-emitting device is installed on the top of the mountain, any remote village below

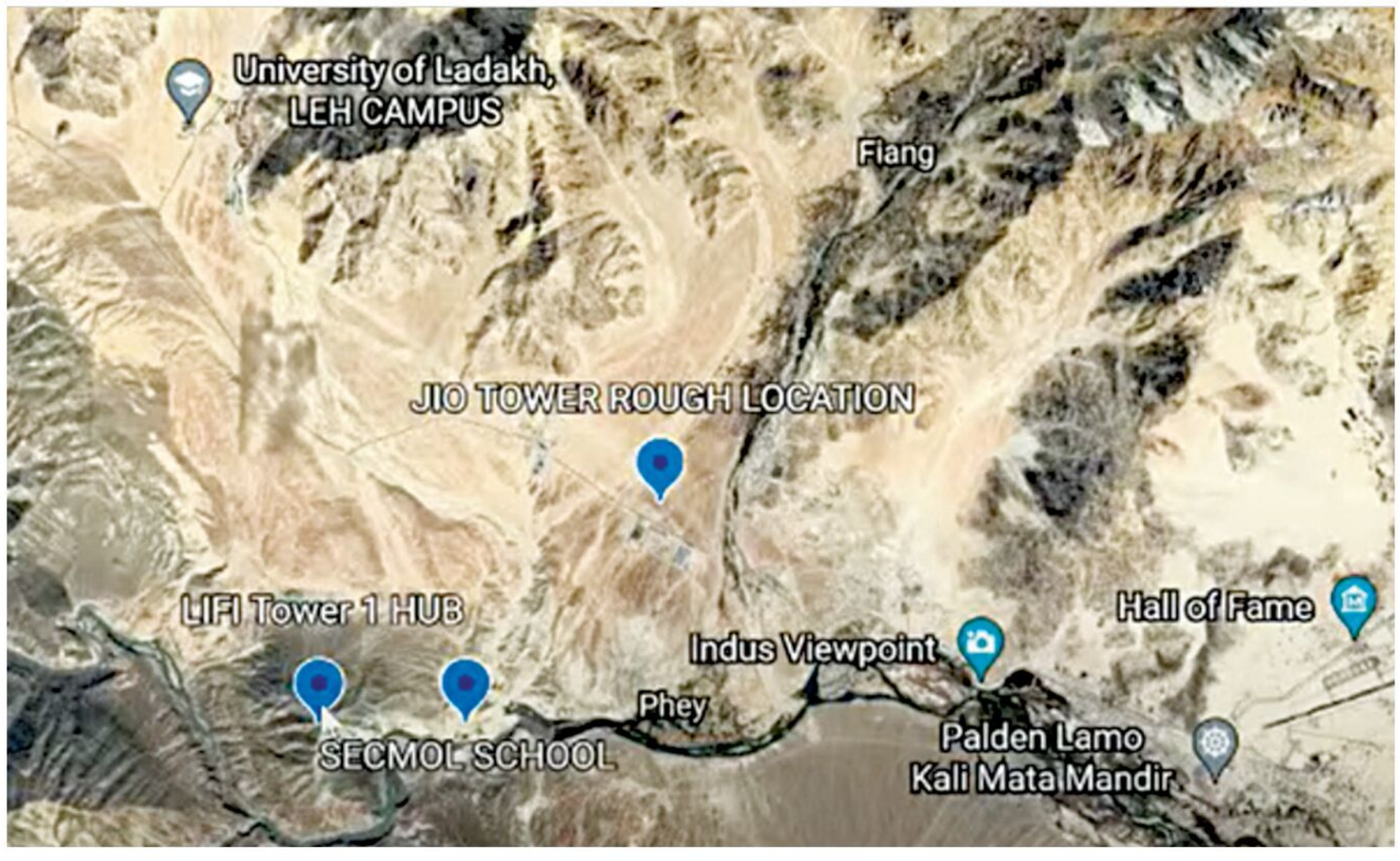


Figure 4: Geographical location

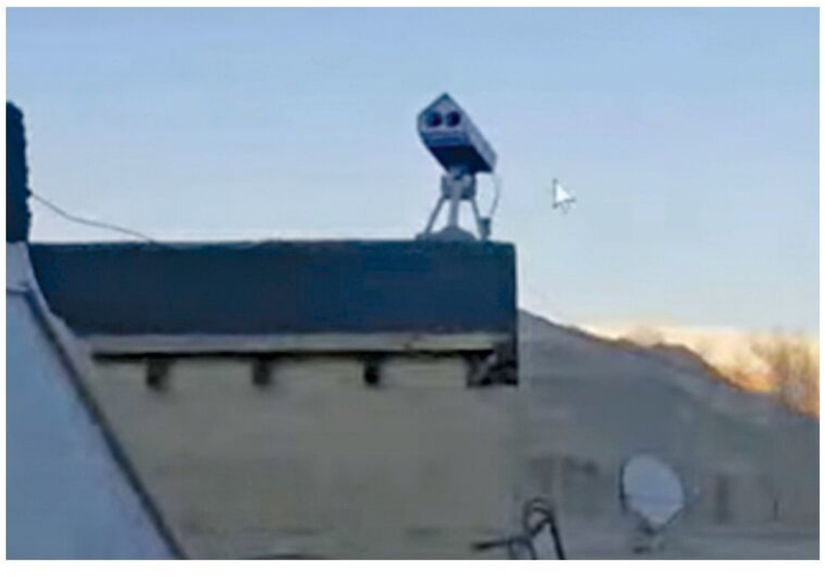


Figure 5: Optical communication unit for Li-Fi

Mountain-to-mountain connectivity

The top of a mountain is usually isolated and cut-off with no internet connectivity found. But mountains are the mightiest and biggest structures that can be used to erect towers. Combining this with an interesting technology called Li-Fi, the mountain-to-mountain connectivity can be achieved.

> "The concept of converting mountains to towers and light rays to internet connectivity for remote areas is something to share with people who are working with electronics and technology.
>
> "When we suggested mountain-tops as towers, normal telecom companies asked how these would be powered and who would maintain them? Since Li-Fi requires very little power, we made power systems that can be carried and are training an army of shepherds in Li-Fi rather than acquiring professionals from other places."
>
> ***—Sonam Wangchuk***

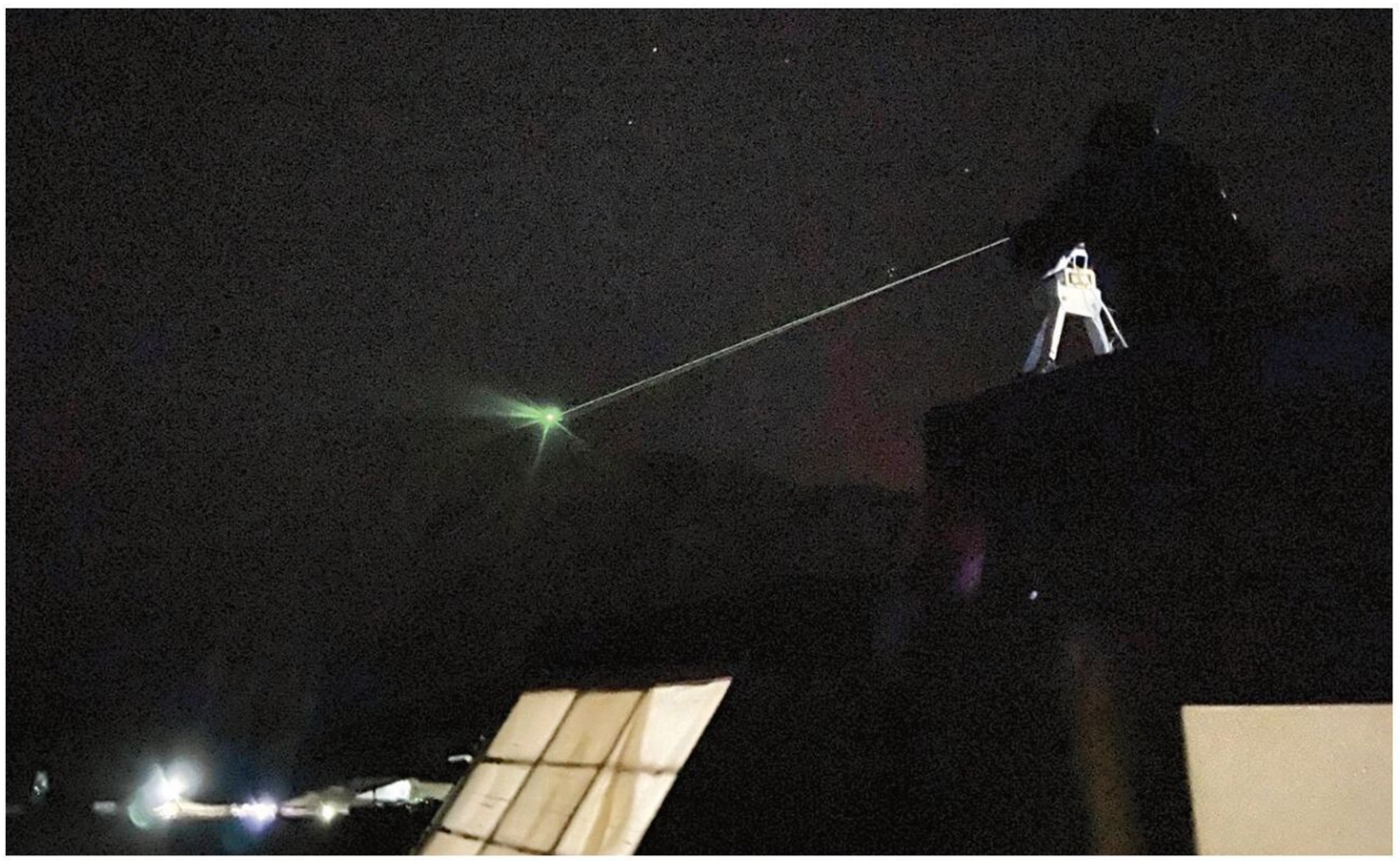


Figure 6: Mountain-to-mountain internet connectivity



Figure 7: Solar-heated battery storage device

the neighbouring mountains will have internet connectivity, enabling students to have online classes. "Using this solution, a lot of cost can be saved," opines Sonam.

The devices used in these innovations are all made in India, in accordance with *Atmanirbhar Bharat*, with the battery and power system being made in the campus itself. Laser beams have a range of around 10km, but in the thin air at high altitudes the range increases up to 15km (still under testing).

Considering this to be a good distance from one mountain to another, it can also be relayed as an optical fibre cable and used for radio frequency for the valleys below where there is population. The capacity of the laser beam transmission (up to 1GB) will replace digging, power, and expenditure for the hilly areas. The speed remains the same in both optical fibre and laser beam communication.

Considering weather conditions, even though there is rain/snow on rare days, the autonomy of the power system is such that even on cloudy days, photovoltaics would generate sufficient electricity (may be only 30 to 40% of their peak), which would help get through the situation.

This technology is now finding its way with major telecom providers and being used for mountain regions. Hence using Li-Fi receivers and transmitters, signals can be obtained from a normal tower to connect remote villages. Telecom providers need to look into this technology that can help solve many problems.

"Using this solution, a lot of cost can be saved," says Sonam. In conclusion, he shares, "These are some examples of just to say that indigenisation and localisation of making things in India is truly a proud moment for the country." END

This article is based on the speech given recently by Sonam Wangchuk at the 35th Year VLSI Conference. The article has been transcribed and curated by EFY's Sharon Abhignya Katta, who is passionate about innovation, especially in the field of telecommunications.

By: Sonam Wangchuk

The author is a resident of Ladakh and winner of the award for Exceptional Contribution in the Areas of VLSI and Embedded Systems

The article was originally published in the September 2022 issue of Electronics For You.

Why Private 5G Services are Needed for Implementing Industry 4.0

5G has arrived! Manufacturers would do well to avail the high bandwidth it brings with it to ensure smoother operations as they embrace Industry 4.0.



With manufacturers opting for Industry 4.0 in order to increase productivity, there is a critical need for reliable connectivity that supports high bandwidth and low latency at the same time.

A Wi-Fi network is most suitable for the indoors but is not scalable to a 10 square kilometre area. To do that we need to install a Wi-Fi modem at every 10 to 20 metres, which is impractical and costly. A cellular network is the most appropriate option here. We already have working 4G networks. These networks are predominantly public networks and not private networks with an enterprise. Similarly, the cloud is also not physically present within an enterprise.

The location of the data center is uncertain. Even if we are aware of its location, the fact that it is travelling through the web network rather than the public network prevents you from knowing how much latency it will incur. This is why this latency needs to be changed. With the introduction of 5G this can be dealt with easily, as it allows you to create an alternative Wi-Fi network within the enterprise. It ensures low latency and high throughput as well as privacy.

Along with that, you also need to have a miniature version of the cloud, which we call the 'H' cloud. It is more reliable for things like augmented reality (AR) and virtual reality (VR), video analytics, robotic operations and autonomous vehicles, and guarantees low latency. It is also important to note that the cloud is essentially for book-keeping and data management and not for any computation tasks.

Information technology applications such as Zoom, as well as system applications and products (SAP) can be run properly on Wi-Fi but operational applications cannot.



Asia's #1 Open Source Conference

Total registrations

10,340

Total attendees

3732

Speakers

90+







www.OpenSourceIndia.in

An EFYGROUP EVENT

19th Edition

OPEN SOURCE INDIA

Thank You

VISITORS, for making the event a huge success

SPEAKERS, for contributing your valuable thoughts

PARTNERS, for your support

Microsoft Azure · Shell · soda foundation · Google Cloud · aws · GitHub · PURESTORAGE · ThriveDX SaaS · harness · DELL Technologies · MOSIP · YML · mt-b · APPLIED SINGULARITY · JAVASCRIPT MEETUP (BANGALORE, INDIA) · OPEN SOURCE BANGALORE Meetup · iotncr · compossible umwelten · APACHE · Apache DevLake · openinventionnetwork · SODA FOUNDATION INDIA MEETUP

For more details, call on +91-98111-55335 or email us at info@opensourceindia.in

Industry 4.0 use cases

Logistics

Warehouses tend to be extremely huge; hence, there is a need to digitise them. Real-time tracking of products across the warehouse is extremely important. Drones can be used for tracking while robotic automation can assist with moving goods from one place to another. AR/VR based applications can be used to assist the training of employees in real time.

Mining

Mining in India is of two kinds. The first is called open pit mining. This kind of mining does not require you to get into a tunnel. The other kind of mining is called closed pit mining. This requires you to go underground inside a tunnel. There can be around 10 to 22 tunnels, and each may be about one or two kilometres long. In places like Australia a lot of automation has been done already while in India things are still done manually. Miners have to enter closed pits and use a USB stick to take the data out from the sensors installed in that area. Only then can they connect it to their office systems and servers in order to use it. This only creates an opportunity to digitise everything. By using automated unmanned vehicles the problem of data extraction can be solved, as no human will need to enter these mines thus significantly reducing the risk of any kind of disaster.

Oil and gas

This is divided into three parts, namely, upstream – offshore/onshore, midstream - pipeline, downstream - refinery. Oil exploration is done both offshore and onshore. This exploration requires a lot of systematic data collection. Heavy analytics is required in order to find out the right places to conduct these explorations. Private 5G can play an important role here, helping increase productivity.

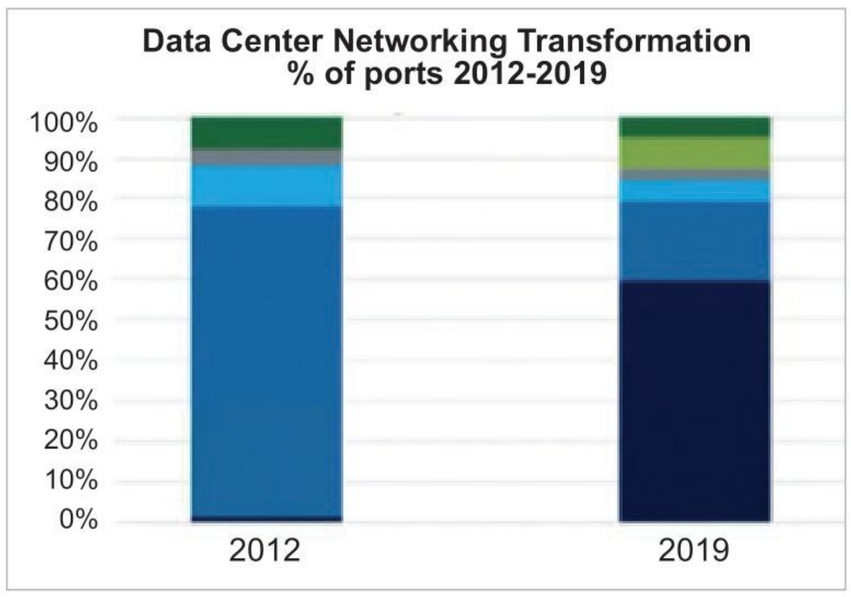


Figure 1: Data center networking transformation

Manufacturing

The Japanese company Fujitsu is making use of the private 5G network as well as edge computing for autonomous vehicles where 4K cameras are being used to ensure that these vehicles are moving properly. Data collected from these 4K cameras as well as sensors allows an analysis of their real-time movement. Similarly, cameras are installed in the manufacturing assembly lines to ensure work is being done the right way. The company uses artificial intelligence (AI) analytics tools to monitor everything. If someone is not doing the work correctly, these AI tools give a proper feedback about it. Training is also being done on the go with the help of AI in real-time.

Openness and disaggregation driven by hyperscalers

Data centres of companies like Facebook and Google have moved to enable openness and disaggregation, especially in the networking space. Back in 2012, switches, routers and other network appliances were all properties owned by the original equipment manufacturers (OEMs). They wanted to run an operating system on multiple pieces of hardware, which was not possible at all. Some hyperscalers like Facebook and Google started using white box hardware, which can be commercially used right off-the-shelf. These are servers that are based on open hardware design. They can be procured easily, and any open software can be put in them. Open software can be taken from various vendors and can run on the white box made by any of the multiple companies. You can even change the software later if you need to, and still be able to run it on the hardware. This is also called disaggregation.

The movement from proprietary solutions to open and disaggregated software and hardware in the data center started in 2012 (refer to Figure 1).

The use of white box hardware was very miniscule at about 2 per cent of the market in 2012. But by 2019, the white box dominated the market at around 59 per cent. This was an eye opener for the telecom industry and a lot of enterprises too. Since then, disaggregation and openness have become effective standards in the data center market. Interestingly, even big companies like Cisco are moving towards disaggregation because that is where the market is headed.

Telecom companies saw the revolution taking place in the data centers, and they knew something similar would drive hardware and software in telecom, predominantly driven by 5G.

Initially, the radio access network (RAN) was proprietary but with the introduction of open radio access network (O-RAN) it has been divided into three parts. The first is the radio unit (RU) which acts like an antenna. Then there is a distributed unit (DU) and a centralised unit (CU). Now, instead of making use of the hardware, you can simply use the cloud to run DU and CU.

The important components of 5G are the radio network and the transport network; the latter sends radio signals to the centralised core. The cores, too, have got disaggregated. There is a transport network between the 5G network and the centralised cloud using a router, which is also getting disaggregated.

At Niral Networks, we have done some indigenous end-to-end integration with a radio company in India, so that we can provide an end-to-end 5G system. Here's a description of our open and disaggregated network operating system called NiralOS.

NiralOS network operating system (NOS)

- *Private 5G core cloud:* Native private 5G core software for mobility, authentication, security, session and policy management. Niral 5G core also has a compact user plane function (UPF) to provide local breakout within an enterprise when integrated with TSP's centralised 5G core.
- *Mobile edge cloud:* Kubernetes and virtualised edge cloud infrastructure to create a mobile edge cloud (MEC) within an enterprise with open APIs to host third-party applications like AR/VR, robotics, drones, AI/ML, video analytics for low latency and privacy.
- *Centralised controller:* Provides centralised management, orchestration, zero touch provisioning and monitoring of multiple private 5G networks and MEC at various sites. The controller can be hosted in the public cloud to centrally manage and monitor multiple private 5G networks.

NiralOS specifications

- Release16-compliant 5G core for private 5G deployment
- 5G network functions--UPF, AMF, SMF, AUSF, UDM, NRF
- Kubernetes based cloud-native network function
- DPDK+VPP based UPF acceleration-linear scaling per CPU core
- UPF local breakout feature for easy integration to TSP's centralised 5G core
- Support of N1, N2, N3, N4 and service based interfaces for 5G SBA
- 5G core deployed on COTS hardware of various form factors and integrated with 5G radio
- Kubernetes and virtualised, cloud agnostic edge cloud to host third-party applications
- Open APIs for integration of third-party application to Niral 5G core and edge platforms
- Web based dashboard for subscriber provisioning configuration and management

Niral Networks has also been awarded for innovation in 5G. We look forward to achieving more such feats. END

By: Abhijit Chaudhary

The author is founder and CEO of Niral Networks. This article is put together from a tech talk he gave at Open Source Conference 2022, an event organised by Samsung R&D Institute India – Bangalore, and IEEE ComSoc, Bangalore Chapter.

Curated and transcribed by: Laveesh Kocher

FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK

We Value Your Feedback

We love to hear from you as **OSFY** consistently strives to make its content **informative, interesting** and **engaging.**



Please share your feedback/ thoughts/ views via email at osfyedit@efyindia.com.

We welcome your comments/ suggestions and encourage you to send them to:
The Editor, D-87/1, Okhla Industrial Area, Phase 1, New Delhi-110020.

OpenSource For You
THE COMPLETE MAGAZINE ON OPEN SOURCE

FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK

The Connect Between Open Source and the Metaverse

This article looks at the relationship between the metaverse and open source. It outlines a few key areas where open source is playing a pivotal role in realising the full potential of the metaverse. It also explores the opportunity that this iteration of the internet and associated virtual spaces presents to reinvent open source.



The metaverse is loosely defined as “a combination of the virtual reality and mixed reality worlds accessed through a browser or a headset, which allows people to have real time interactions and experiences across distance.” It is generating a lot of attention now, as it matures beyond its previous predominant realm of gaming into its much broader field of application as a place not only to play, but also to socialise and work.

Both ‘metaverse’ and ‘open source’ are catch-all phrases in their relative fields, covering everything from technology and communities, to markets, methodologies, and more.

A considerable number of components of the metaverse are being developed as open source projects, according to open source principles — for example, self-sovereign identity (SSI) and smart contracts, both of which are built on blockchain technology.

Interoperability and connectedness

One of the reasons the internet has been so lucrative and transformative is because of how it was created. Throughout the 1960s to 1990s, the foundation of today’s internet was built through a variety of consortiums and informal working groups composed of government research labs, public

universities, and independent technologists. These usually not-for-profit collectives typically focused on establishing open standards that would help them share information from one server to another (i.e., messages or files), and in doing so make it easier to collaborate on future technologies, projects, and ideas.

Many believe that the definition and success of a metaverse requires it to be a heavily decentralised platform built mostly upon community-based standards and protocols like the open web and an 'open source' metaverse operating system or platform. However, this does not mean there will not be dominant closed platforms in the metaverse.

Interoperability and connectedness are believed to be two of the top core attributes of the metaverse. Interoperability, as per *TechTarget*, is the ability of different systems, devices, applications, or products to connect and communicate in a coordinated way, without effort from the end user.

In the metaverse, interoperability is the ability for different virtual worlds and platforms to interact with each other. It means that users on one platform can communicate and interact with users on another platform, as well as share data and content. If we have a lot of virtual worlds where you cannot go from one to the next or they are incompatible with each other, that is not going to be a pleasant experience. So, the very nature of a connected system requires that there be openness.

When operators build their metaverse, one can anticipate a constant battle across the technology stack between centralised and proprietary versus decentralised and open standards and protocols. It will benefit users

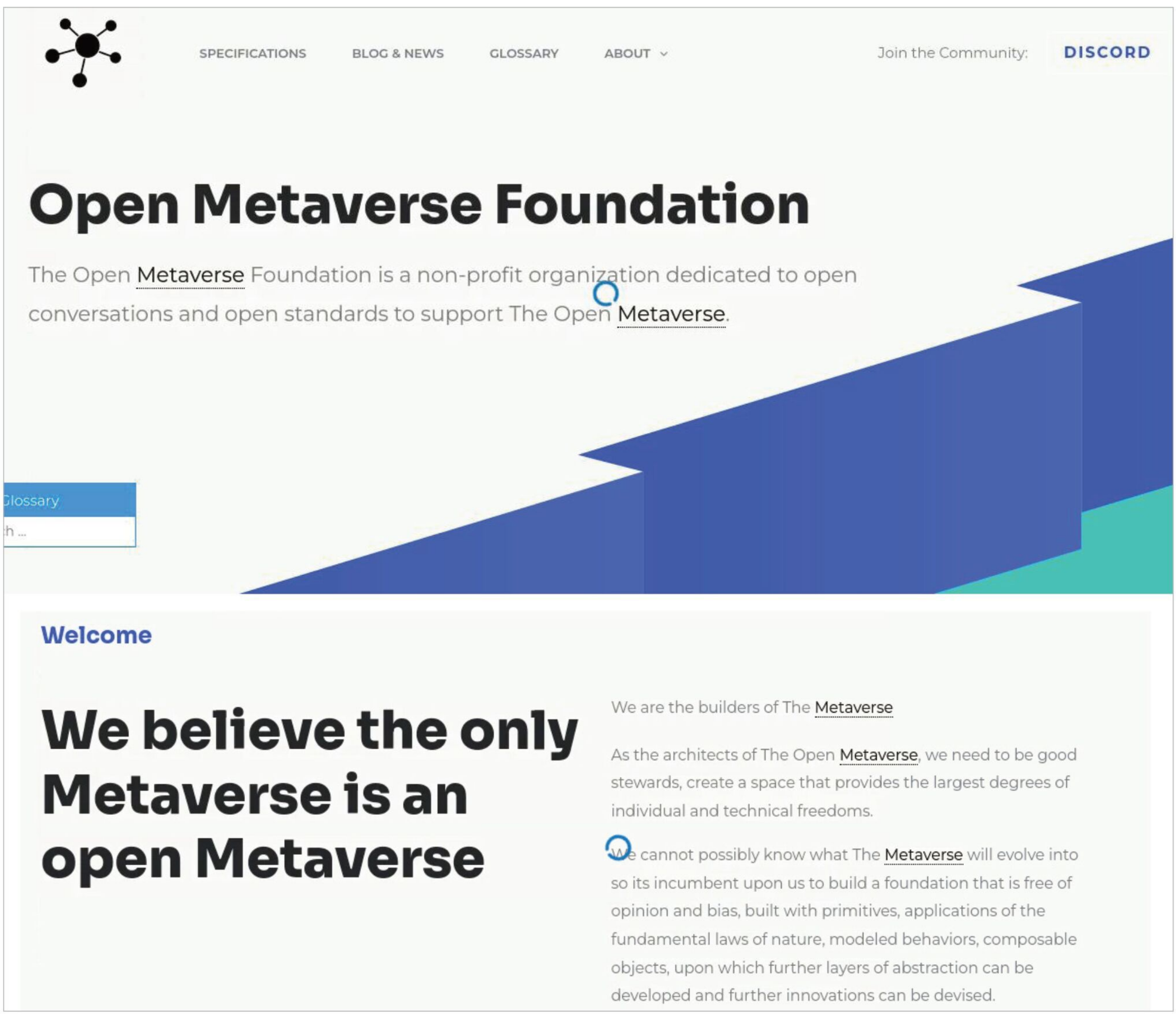


Figure 1: Home page of the Open Metaverse Foundation

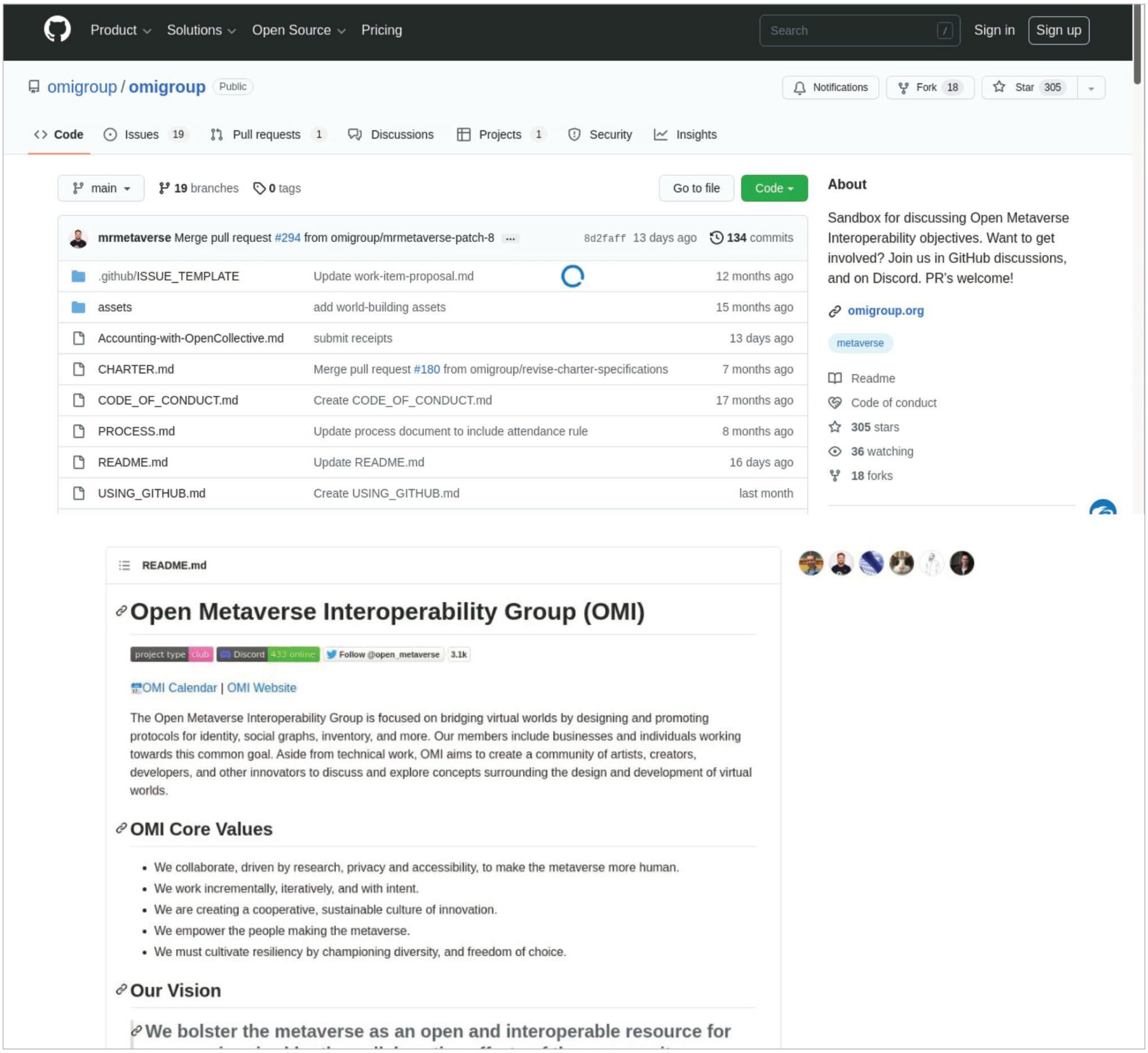


Figure 2: GitHub repo of the Open Metaverse Interoperability (OMI) Group

if the metaverse builders collaborate on interoperable, open standards on privacy, security, blockchain, data storage, AI (artificial intelligence), spatial computing, and so on.

For an open, interoperable metaverse to see its potential, it is vital for consortiums of different enterprises, both centralised and decentralised, to collaborate on protocol standards. Much as the internet's globally accepted standards of DNS (domain name system) and IP (internet protocol) catalysed its adoption, the metaverse requires standards for graphical, transactional, compute, and network capabilities.

Metaverse open standards

In the initial stages of development, there are already some projects working on creating an interoperable metaverse. One of them is the Open Metaverse Foundation, which is creating an open source platform that allows different virtual worlds to interoperate with each other.

There are some efforts to create standards for the metaverse, but these are in their infancy. W3C, the consortium that standardised web navigation, created the Open Metaverse Interoperability (OMI) Group in April 2021, which is focused on bridging virtual worlds by designing and promoting protocols for identity, social graphs, inventory, and more.

Another such initiative is the Metaverse Standards Forum launched recently by the big tech companies of



Figure 3: Home page of the Metaverse Standards Forum

the world, namely, Meta, MSFT, Epic, and 33 others.

The forum will explore where the lack of interoperability is holding back metaverse deployment and how the work of Standards Developing Organisations (SDOs) defining and evolving needed standards may be coordinated and accelerated. The forum will not create standards *per se*. Instead, it will take on a more top-level, one might say 'meta', role by coordinating requirements and resources to foster the creation and evolution of standards for metaverse within standards organisations working in relevant domains.

Metaverse technologies and open source metaverse applications have predominantly been focused within the gaming space. However, as the concept, technologies and supporting infrastructure advance in concert with a general increase in public consciousness of the metaverse, we are seeing a rapid increase of metaverse based services and applications. This includes applications for business, where there are numerous opportunities to rethink user interaction, distribution, and revenue channels, as well as how (and where) work is orchestrated and conducted.

There are several neighbouring yet distinct technologies that will need to be integrated and adopted into the metaverse for this to be fully realised. Amongst the most important of these are blockchain technologies, cloud infrastructure, Internet of Things (IoT), artificial intelligence, machine learning, robotics, and connectivity-based infrastructure such as satellite internet constellations and 5G/6G — all built on or predominantly open source.

Open source holds a critical role in the development of metaverse-aligned technologies and establishing the standards that will underpin the technology. There are numerous macro challenges that present significant obstacles to wider adoption of the metaverse -- for example, interoperability, identity, privacy, security, sustainability, governance, and inclusivity. Open source software, projects and communities are essential for addressing and agreeing upon the best way to satisfy these challenges.

Blockchain technology will play an extremely vital role in surmounting these challenges and, consequently, many of the associated projects and communities of practice are open source. A prime example of this surrounds the projects focused on self-sovereign identity (SSI), which is an approach that gives individuals control of their digital identities and addresses the difficulty of establishing trust in an interaction. Given the transient and virtual/remote nature of human interaction in the metaverse, this question of identity, ownership and trust is at the heart of whether the metaverse is a safe, inclusive, and mutually beneficial place to work, socialise and play, or the very opposite.

The metaverse is increasingly becoming a viable place to work and to sell products and services, and

where organisations should focus their attention, effort, and resources. The metaverse market is still at the foot of its growth curve and the associated technologies are still in the formative stages. As such, some of the key building blocks are still works in progress.

Much of this work is being conducted in open source communities and projects, and represents a fabulous opportunity for open source oriented organisations to 'get in on the ground floor' and cement their position as thought leaders and pioneers in this exciting and emerging technology.

Metaverse components and open source

Now let us consider some of the key components that will facilitate the ability and scalability of working in the metaverse.

Identity: The metaverse and its future are intrinsically linked to web3, billed as the next iteration of the World Wide Web. It is based on blockchain technology, which incorporates concepts such as decentralisation and token-based economics. The key term here is 'decentralised', i.e., not managed or controlled by a centralised body, or company.

In the present iteration, web2, the market and our data (including our digital identity) is monopolised and monetised by a handful of huge technology companies such as Meta (previously Facebook), Google, Microsoft, etc.

A key tenet of web3 is to liberate, or democratise and decentralise our personal data, including our digital identity. There will, at least in a first phase, not be one single, vast metaverse, but thousands of metaverses.

In the future, some predict that, as there is one internet, there will be one metaverse, interconnecting multiple sub-metaverses in one coherent, standards based, experience. Some metaverses will be private, some will be public, and many will be a combination of the two. This will lead to a highly transient experience, both as consumers and increasingly as workers. And, therefore, SSI is such an exciting and important facilitation component for the metaverse.

SSI will handle standard authentication requirements, but so much besides. It has the potential to support the concept of 'living resumes', which will accurately and exhaustively capture your work, education, and deliverable history. Background verification checks would be made near instantaneous, as all this information could be kept up-to-date and validated through SSI.

Another potential application is pensions, pension contributions and employment benefits, something that will prove increasingly difficult to track as employment tenures continue to shorten.

There is an incredibly significant level of contribution being made by the open source community to SSI. Notable examples are:

- The European Self Sovereign Identity Framework Lab (eSSIF-Lab)
- Hyperledger Foundation (under the Linux Foundation)
- Trust Over IP Foundation (under the Linux Foundation)
- IDUnion
- Sovrin Foundation
- World Wide Web Consortium (W3C)
- Decentralised Identity Foundation (DIF)
- Network of Networks

Smart contracts: Another fundamental application of blockchain technologies that will be a critical component of the metaverse is smart contracts.

There are numerous potential applications of smart contracts that will help resolve usual challenges and areas of contractual inefficiency -- for example, statements of work and payment of workers. Another potentially interesting application is in the eRFx procurement technology market. In procurement technology, eRFx is an acronym for Electronic Request For [x], where x can be Proposal (RFP), Quotation (RFQ), Information (RFI) or Tender (RFT). A huge amount of investment in the typical enterprise is communicated and tendered through this procurement process. By leveraging smart contracts and linking this to the new talent delivery model we can significantly streamline the entire process.

As an application, the smart contract lends itself much more readily to the 'matching' process. It also provides invaluable information to 'shift-left' the visibility of demand skills to the talent community and vendors alike. Fulfilment and payment milestones could be automated because of smart contracts, to name just a few benefits.

Another entity associated with smart contracts is a decentralised autonomous organisation (DAO). A DAO is effectively a group of people who agree to abide by certain rules for a common purpose. Those rules are written into the code of the organisation via smart contracts on the blockchain (algorithms that run when certain criteria are met). Open source projects and their communities require a similar level of coordination and governance.

Given that the next iteration of the internet, web3, is built upon blockchain and is intrinsically linked to the metaverse, there is a potential opportunity to rethink where code is developed and versioned, as well as how to foster collaboration, provide training, and support communities of practice in the open source community. DAOs could be a vehicle to provide the governance layer for that vision.

Digital twins

For the metaverse to become a genuine place for work, it will be necessary to replicate assets ('assets' being anything from a container through to an entire operating system and everything in

between). According to Accenture's *Technology Trends 2022* report, the global digital twin market was valued at US$ 3.21 billion in 2020 and is expected to grow to US$ 184.5 billion by 2030. IBM breaks digital twins into four categories:

- Component/Parts twins
- Asset twins
- System/Unit twins
- Process twins

Microsoft is the frontrunner in this exciting space with its Azure Digital Twins offering, but there are clear opportunities for the open source community to investigate the observability aspect of digital twin technologies.

Listed above are some of the critical components for the metaverse and its growth trajectory that open source is already involved in. There are clearly opportunities for open source oriented organisations to both engage with these open source projects and foundations. That engagement could take the form of thought leadership, evangelism, and training, as well as technical contributions to address some of the persisting challenges surrounding the metaverse such as privacy and security concerns.

However, there is a bigger opportunity that presents itself here. Namely, what could open source software development and communities look like in web3 and the metaverse? The modern technologies and approaches involved offer a unique opportunity to redesign it, to usher in an evolutionary step forward for open source. END

Open source projects in the smart contracts space

- Hyperledger Foundation (under the Linux Foundation)
- Accord Project
- Chainlink
- Waceo

References

- Interchange Tools + Standards and the Metaverse, by Matthew Ball and Jacob Navok, *https://www.matthewball.vc/all/interchangemetaverse*
- Metaverse Standards Forum, *https://metaverse-standards.org/*
- Digital Identity Ecosystems, *https://pcsi.nl/projects/digital-identity-ecosystems/*
- E-procurement, *https://wikimili.com/en/E-procurement*

By: Ankit Pandey

The author is an architect with open source strategy at Lab45 (CTO office) in Wipro Limited.



THE COMPLETE MAGAZINE ON OPEN SOURCE

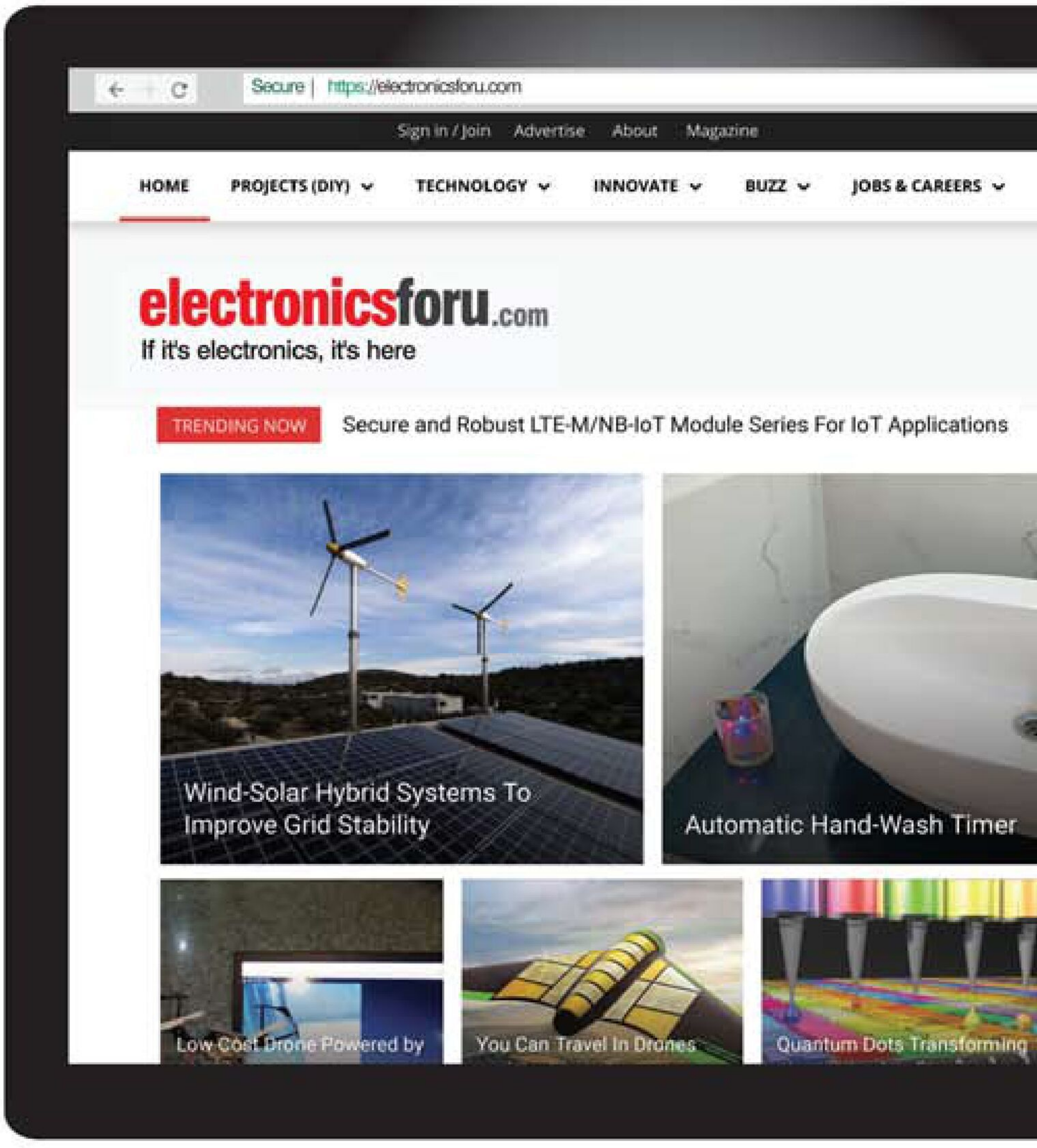


» To find dealers to buy copy from news-stand, visit: www.ezine.lfymag.com

» To subscribe, visit: pay.efyindia.com

» To buy an ezine edition, visit: www.magzter.com choose Open Source For You

Your favourite website has



electronicsforu.com

THANKS TO YOU—OUR ONLINE NETWORK IS

FACTS & FIGURES

- 4 websites (two more coming soon)
- Five major Facebook communities
- Seven major LinkedIN groups & pages
- Million-plus active users (monthly)
- Million-plus reach through Facebook
- Fifty-thousand-plus industry connections through LinkedIN

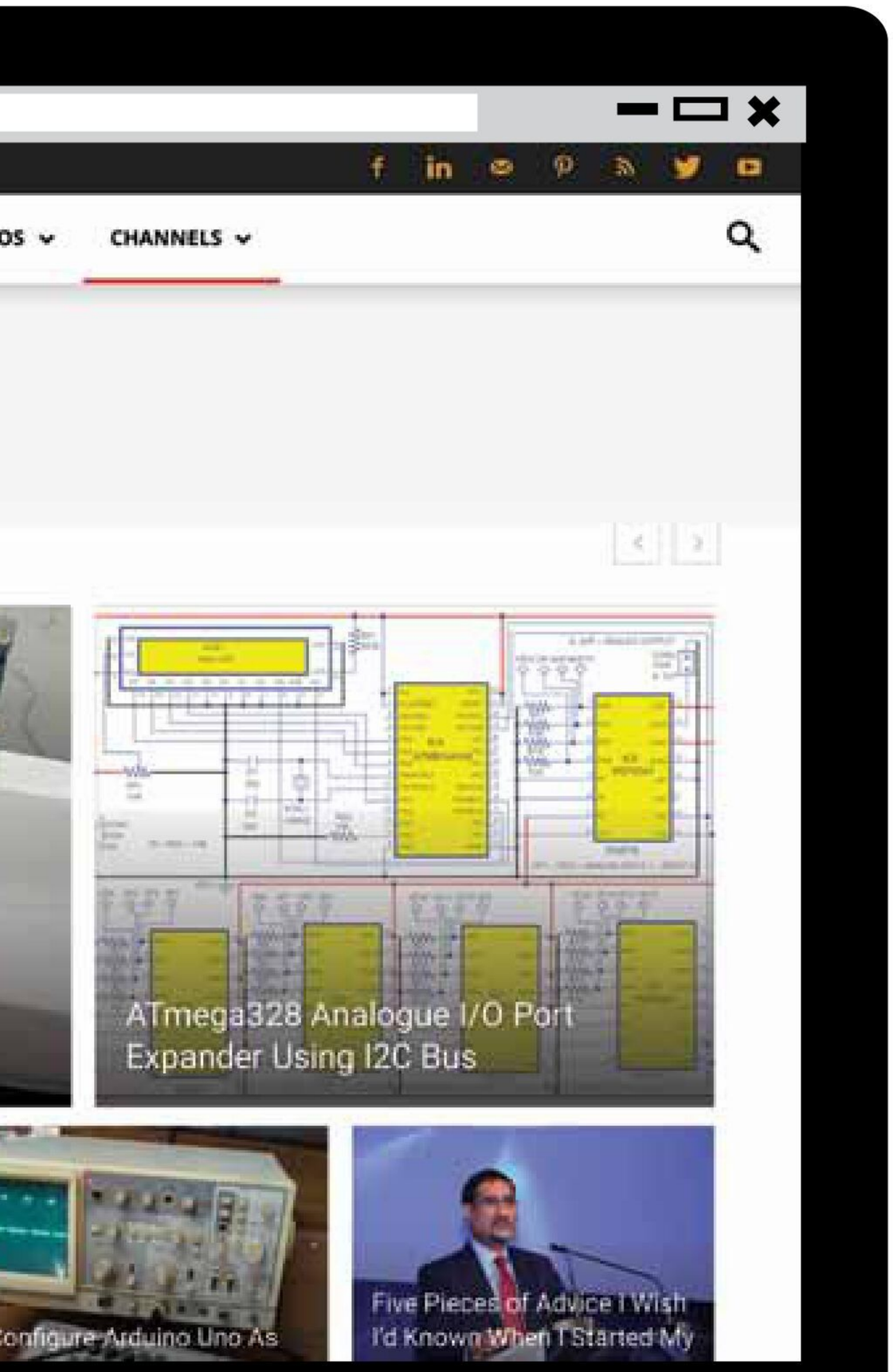
READERS

- You can access all content for FREE
- You can subscribe to newsletters for FREE --on most websites
- Register on our websites to get free invites to technical webinars and seminars

EXPERTS

- Experts who want to share their knowledge through articles, DIY Projects, etc are welcome
- We also welcome experts who want to share their knowledge through webinars or seminars
- You can contact us at editop@efy.in

fast growing peers now...





The Latest in IOT.
A platform for enablers, creators and providers of IOT solutions.



India. Electronics. Directory.
Enabling commerce between buyers & sellers of electronics in India.



Business. Electronics. India.
Everything you want to know about India's electronics industry.

Amazing DIY Projects. Latest Tech trends.

The hang-out for electronics enthusiasts.

AMONGST THE WORLD'S TOP 5 AND GROWING!

INDUSTRY

- You can advertise for as little as US$ 100 per month
- Special combo offers for advertisers in our print publications
- We've now enabled flexible CPM-based advertising
- You can advertise on the platform of your choice (based on your target audience)
- We invite press releases at efy-edit-team@efy.in
- Press releases are published free of cost, subject to discretion of the editorial team

RESPONSE GUARANTEED SOLUTIONS

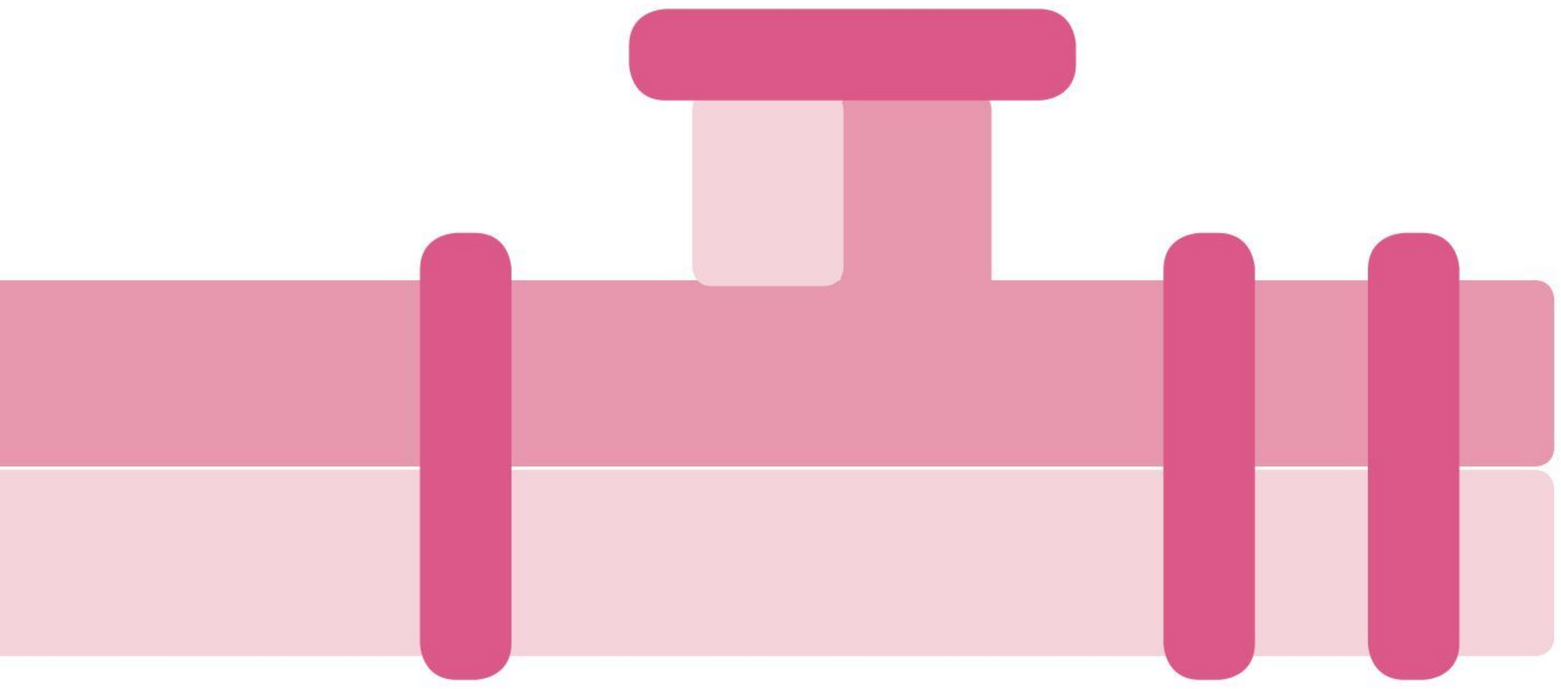
We now also act as marketing partners for our clients and drive entire marketing for them, where we charge them on basis of results and not efforts!

CONTACT US: Shrikant Rao • growmybiz@efy.in • +91-98111 55335

FOCUS

Data Management

Using Apache Beam to Pipeline and Process Data



We live in an era where data is growing at a huge pace and will continue to do so as new technologies evolve. This data needs to be pipelined in order to bemanaged and analysed. Apache Beam helps with that. *Read more on page.....39*

Open Source Tools for Data Monetization: A Quick Look



According to Wikipedia, “Data monetization, a form of monetization, may refer to the act of generating measurable economic benefits from available data sources.” Let us briefly look at the open source tools that can help with this. *Read more on page.....44*

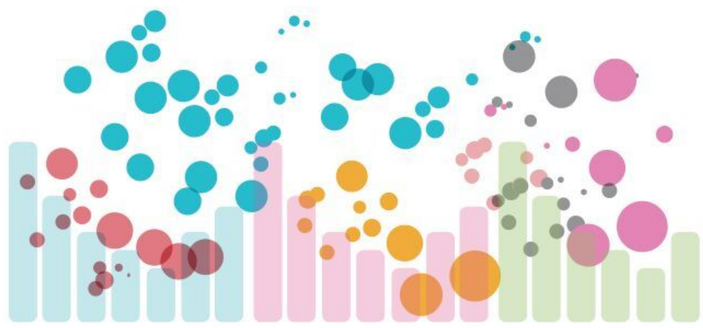
Installing and Setting Up UrBackup on Ubuntu



This is the age of data. Keeping a backup of this data is critical for every business. This short tutorial explains how to install and use UrBackup, a simple open source tool, for data backup.

Read more on page.....46

RAWGraphs: The No-Code Data Visualisation Tool

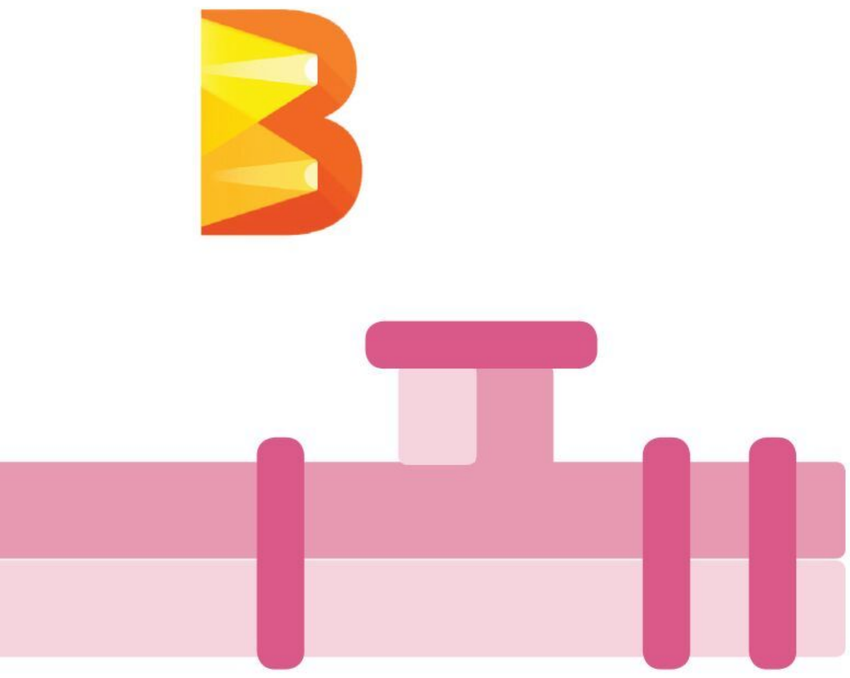


We need specific and effective tools to make sense of the vast amounts of data being generated today. This article looks at an interesting open source data visualisation tool called RAWGraphs, which helps build professional quality visualisation without requiring any code. Let’s see how.

Read more on page.....48

Using Apache Beam to Pipeline and Process Data

We live in an era where data is growing at a huge pace and will continue to do so as new technologies evolve. This data needs to be pipelined in order to be managed and analysed. Apache Beam helps with that.



Across the globe, tons of data is being generated every moment, thanks to:

- Social communication networks
- E-commerce platforms
- Supply chain management
- Stock markets
- Maps/location data
- Fraud detection in banking and finance
- Log files monitoring and analysis
- Electronic health records
- E-governance data sets

To process such data effectively, there is a need to integrate data pipelines so that congestion-free communication can take place.

Data flow pipelines and batch processing

The term data pipeline refers to the effective and rapid process of transporting data from its source to its destination (including the data warehouse). Data undergoes several transformations and optimisations along the road, ultimately arriving in a form suitable for analysis and business insights.

Data pipelining has many stages for real-time applications. Data is first ingested at the start of the pipeline. After that, there is a sequence of operations, each of which produces an output that serves as an input for the subsequent operation. This procedure is repeated until the pipeline is finished. It's possible to execute separate processes simultaneously.

The transmission of data from one system to another is fraught with the potential for errors or slowdowns due to the complexity of the networks involved. The extent and effect of the issues only grow as the breadth and depth of the data's function expands. For this reason, data pipelines are crucial. They allow seamless, automatic transfer of data from one phase to the next, reducing the need for any human intervention. They are crucial for real-time analytics, which allow you to act on information more quickly.

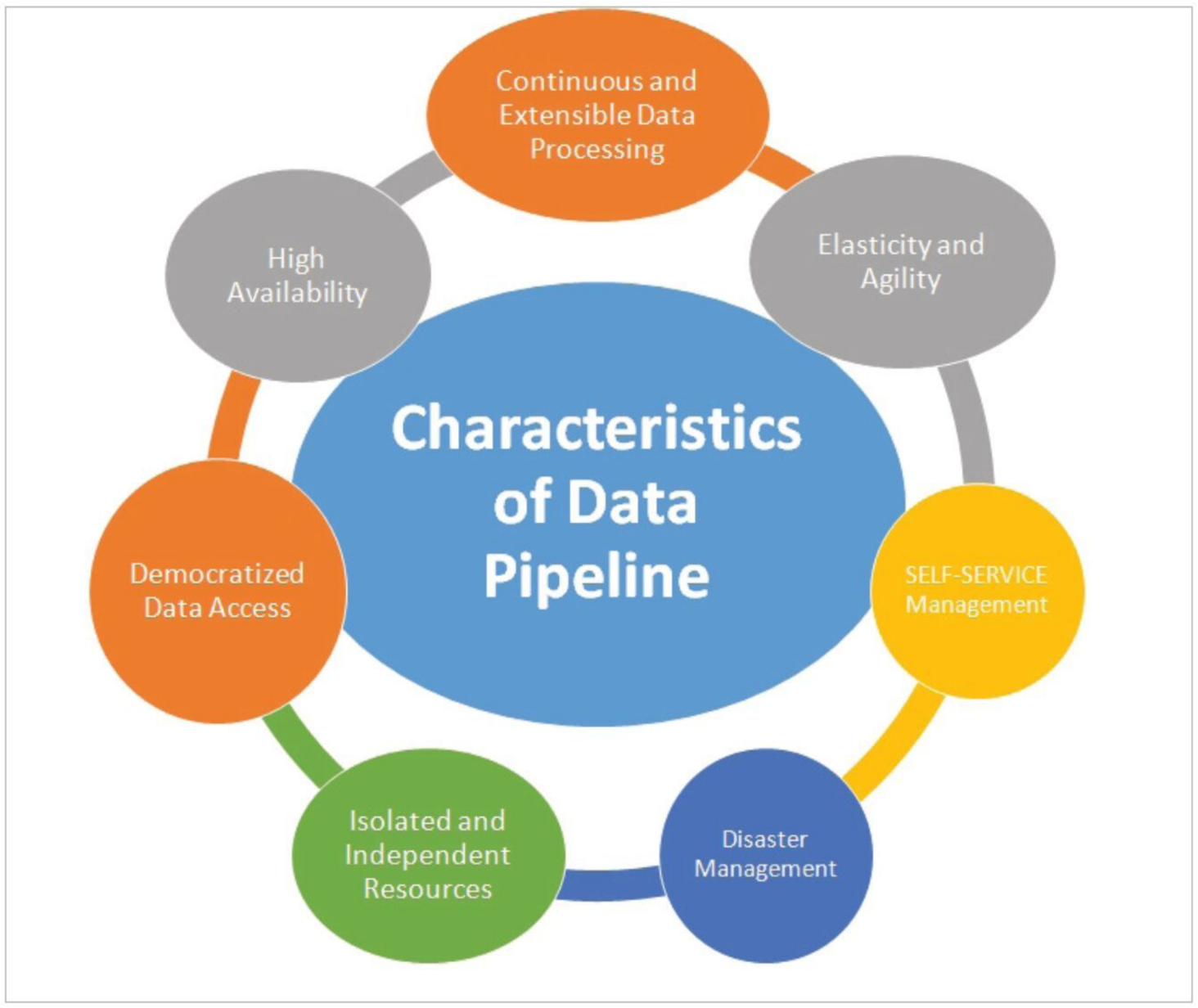


Figure 1: Key characteristics of a data pipeline

Assuring consistent data quality and allowing rapid analysis for business insights can be achieved by merging data from several silos into a single source of truth. Also, information from the point of sale (PoS) system can be processed in real-time using a streaming data pipeline. The stream processing engine may loop the pipeline's outputs back to the PoS system or send them to other applications like data repositories, marketing apps, and customer relationship management (CRM) software.

Each data pipeline has three main parts: a data source, some sort of processing step(s), and a final destination. A data pipeline's last stop may be referred to as a 'sink' in this context. Data pipelines facilitate the transfer of information between several systems, such as an application and a data warehouse, a data lake and an analytics database, or a data lake and a payment processing system. It is possible for a data pipeline to have a single source and a single destination, in which case the pipeline's sole purpose would be to effect changes to the data collection. A data pipeline exists wherever information is sent from point A to point B (or B to C to D).

More and more data is being moved between applications, and as organisations seek to build applications with small code bases that serve a very specific purpose (called 'microservices'), the efficiency of data pipelines has become crucial. Numerous data pipelines may be fed by information produced by a single system or application, and each data pipeline may in turn feed multiple other pipelines or applications.

Table 1: Prominent software suites for data streaming and analytics

Software suite	URL
KNIME Analytics Platform	*https://www.knime.com/knime-analytics-platform*
RapidMiner	*https://rapidminer.com/*
Apache Spark	*https://spark.apache.org/*
Pentaho	*https://www.hitachivantara.com/en-us/products/lumada-dataops/data-integration-analytics/download-pentaho.html*
Kafka Streams	*https://kafka.apache.org/*
Atlas.ti	*https://atlasti.com/*
Apache Storm	*https://storm.apache.org/*
Apache Hadoop	*https://hadoop.apache.org/releases.html*
Apache Cassandra	*http://cassandra.apache.org/download/*
Apache Flink	*https://flink.apache.org/*
OpenRefine	*https://openrefine.org/*
Data Cleaner	*https://github.com/datacleaner*
Apache Nifi	*https://nifi.apache.org/*

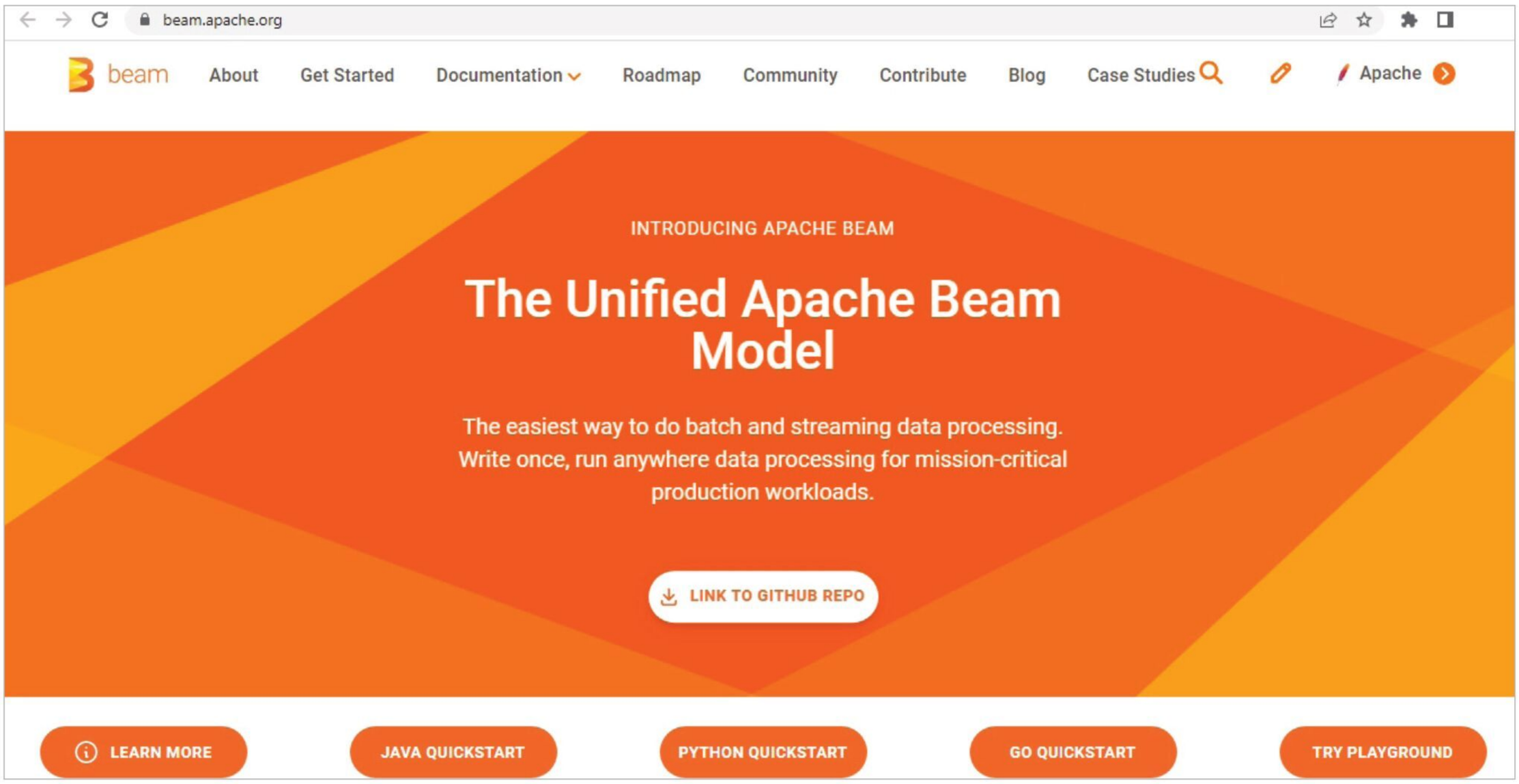


Figure 2: Official portal of Apache Beam

www.opensourceforu.com

Job Guaranteed.

You don't need a training institute. Just do it yourself.

WORK WITH OPEN SOURCE.

Demand for Open Source is sky rocketing. Be it for managing IT infrastructure or development of software —Open Source experts are what organisations are seeking. Develop expertise in an Open Source stack, and watch your career graph zoom.

You don't need a training institute for this. You can do it yourself. And, *Open Source For You* can be your friend and a guide through this journey.

TO READ OUR PRINT EDITION Visit: https://subscribe.efyindia.com

TO READ OUR EZINE EDITION Visit: https://ezine.lfymag.com

WORLD'S LEADING PUBLICATION ON OPEN SOURCE

Looking for marketing solutions to engage with cutting edge techies?
Contact us at growmybiz@efy.in OR call us at +91-9811155335.



Let's take an example of the impact of a single social media post. A real-time report that tracks social media mentions; a sentiment analysis tool that gives a good, negative, or neutral conclusion; or an app charting each mention on a globe map might all be fed with data from this event. Despite sharing a common data source, these apps all rely on different data pipelines that must run without a hitch to produce the desired output for the end user.

Data transformation, enhancement, filtering, grouping, aggregation, and the application of algorithms to data are all typical components of data pipelines.

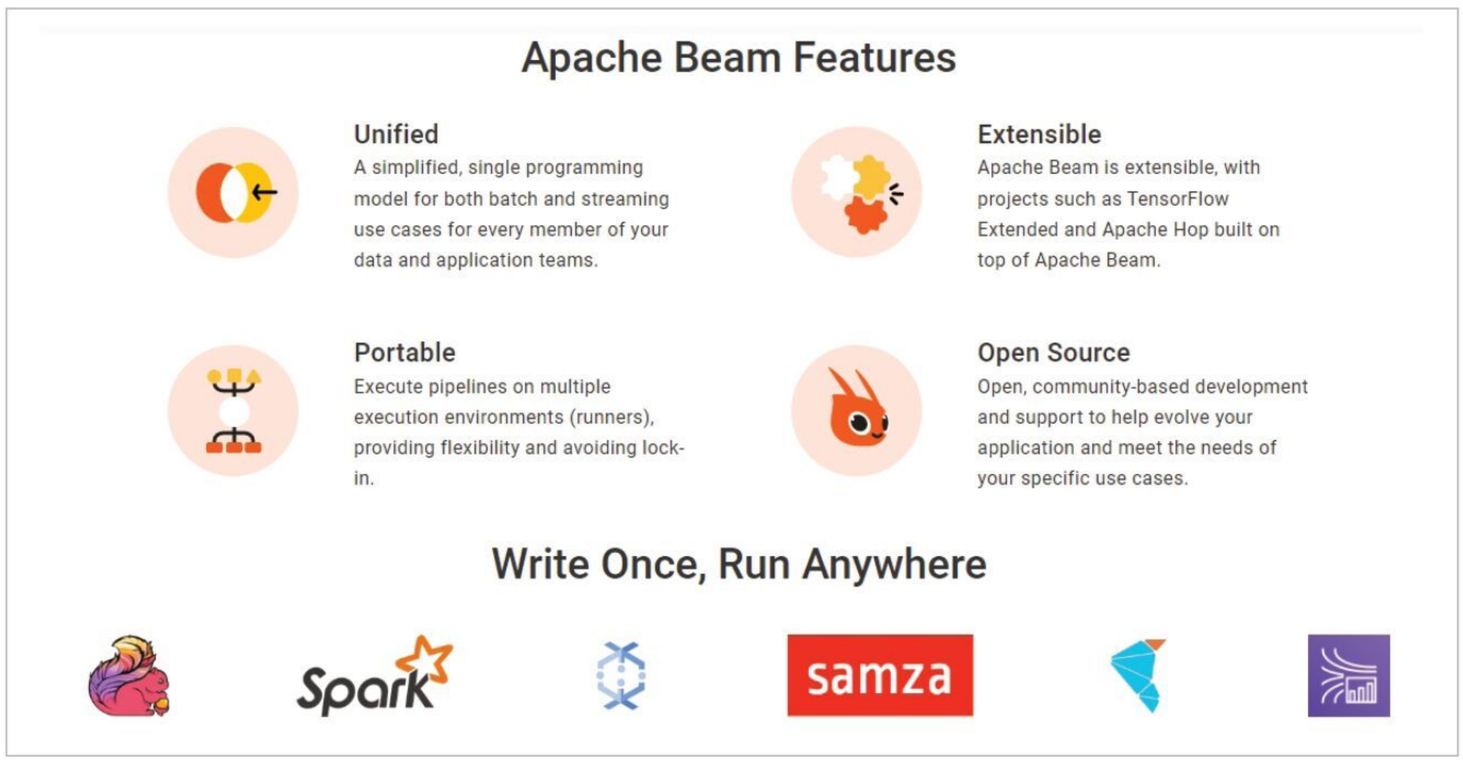


Figure 3: Key features and integration libraries with Apache Beam

Pipelines in Big Data scenarios

The term Big Data indicates that there is a large amount of data to process, and this is true because the volume, variety, and velocity of data have all increased considerably in recent years. Having access to all this information can be useful for a wide variety of purposes.

Data pipelines have progressed with other parts of data architecture to accommodate huge data. A compelling reason to construct streaming data pipelines for Big Data is the volume and speed of this data. These pipelines help to gather information in real-time and act upon it.

The Big Data pipeline needs to be scalable in order to process massive amounts of data concurrently, as there are likely to be numerous data events that occur at the same time or very near together. The pipeline must also be able to detect and process heterogeneous data, including structured, unstructured, and semi-structured data.

Working with Apache Beam

Apache Beam (*https://beam.apache.org/*) is a free and open source unified programming model for describing and working with effective data processing pipelines. This data processing methodology is used to work with the pipelining and real-time filtering of data for multiple applications. All data sets and data frames can be processed using a single API with the integration of Apache Beam. To 'build and run in any place' is a key concept of Apache Beam.

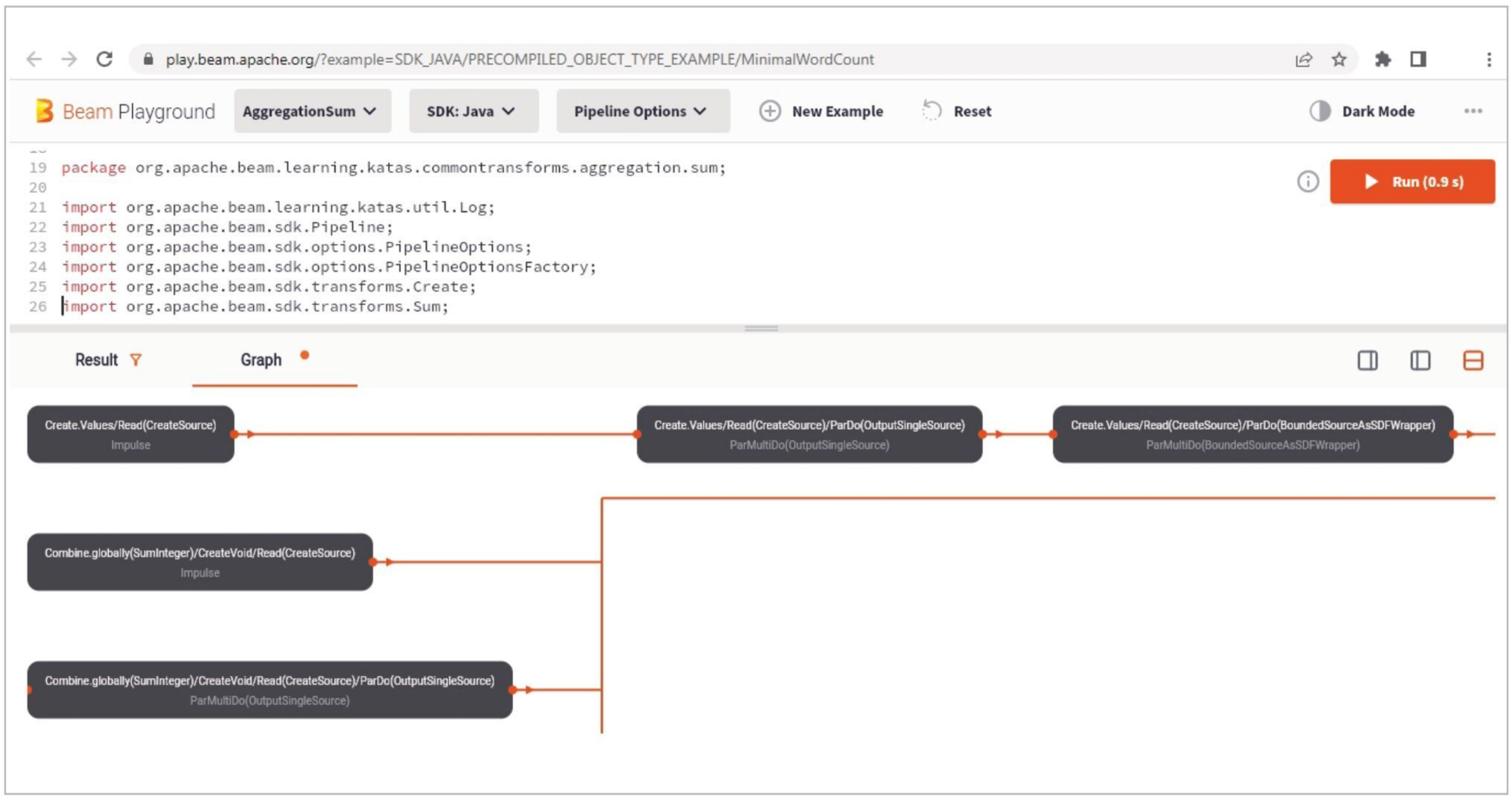


Figure 4: Apache Beam Playground

Apache Beam reduces the complexity of processing massive amounts of data. It is widely used by thousands of businesses across the world thanks to its innovative data processing tools, proven scalability, and robust but scalable features.

The key features of Apache Beam are:

- Powerful abstraction
- Unified batch and streaming programming model
- Cross-language capabilities
- Portability
- Extensibility
- Flexibility
- Ease of adoption

Apache Beam is also available in its cloud based variant called Apache Beam Playground *(https://play.beam.apache.org/)*, with the help of which programs can be executed online (Figure 4) using a range of programming languages including Java, Python, Go and SCIO.

Installation of Apache Beam on a dedicated server

Apache Beam can be installed on a dedicated server or Google Colab, as follows:

```
! pip   install apache-beam
```

To work with data pipelines, it uses the following segments.

- *Pipeline:* Reading, Processing, Storage of data
- *PCollection:* Data frames, Accessing data sets
- *PTransform:* Data transformation, Windowing, Watermarks, Time stamps
- *Runner:* Operation of the pipelines

Here is a scenario where a data pipeline is executed on a CSV movies data set that has the following attributes:

1. movie-id
2. type-of-show (movie/web series)
3. release-year
4. duration-of-the-movie

The following code can be applied to filter and fetch the records matching 'Movie':

```
import apache_beam as apachebeam
datapipeline = apachebeam.Pipeline()
moviesdataset = (
  datapipeline
    | apachebeam.
io.ReadFromText("moviesdataset.csv",
skip_header_lines=1)
    | apachebeam.Map(lambda line:line.
split(","))
    | apachebeam.Filter(lambda
line:line[1] == "Movie")
    | apachebeam.
io.WriteToText("results2.txt")
  )
datapipeline.run()
```

results.txt-00000-of-00001

```
1  s1,TV Show,2020,TV-MA,4 Seasons
2  s2,Movie,2016,TV-MA,93 min
3  s3,Movie,2011,R,78 min
4  s4,Movie,2009,PG-13,80 min
5  s5,Movie,2008,PG-13,123 min
6  s6,TV Show,2016,TV-MA,1 Season
7  s7,Movie,2019,TV-MA,95 min
8  s8,Movie,1997,R,119 min
9  s9,Movie,2019,TV-14,118 min
10 s10,Movie,2008,TV-MA,143 min
11 s11,Movie,2017,TV-MA,103 min
12 s12,TV Show,2018,TV-MA,1 Season
13 s13,TV Show,2019,TV-MA,1 Season
14 s14,Movie,2018,TV-MA,89 min
15 s15,Movie,2019,R,91 min
16 s16,Movie,2014,TV-14,149 min
17 s17,TV Show,2018,TV-14,1 Season
18 s18,Movie,2018,R,144 min
```

Figure 5: Filtered results from Apache Beam

Pipeline operations on a large data set can be implemented with minimum lines of code using Apache Beam.

Modern data engineering scenarios make use of Snowflake platforms that have enhanced structured formats for implementing data pipelines. Snowflake's data pipelines are flexible, allowing for either batch or continuous processing. Data transformation and optimisation for continuous data loading are typically labour-intensive tasks; however, modern data pipelines automate many of these processes END

By: Dr Gaurav Kumar

The author is associated with various academic and research institutes for delivering expert lectures and conducting technical workshops on the latest technologies and tools.

Open Source Tools for Data Monetization: A Quick Look

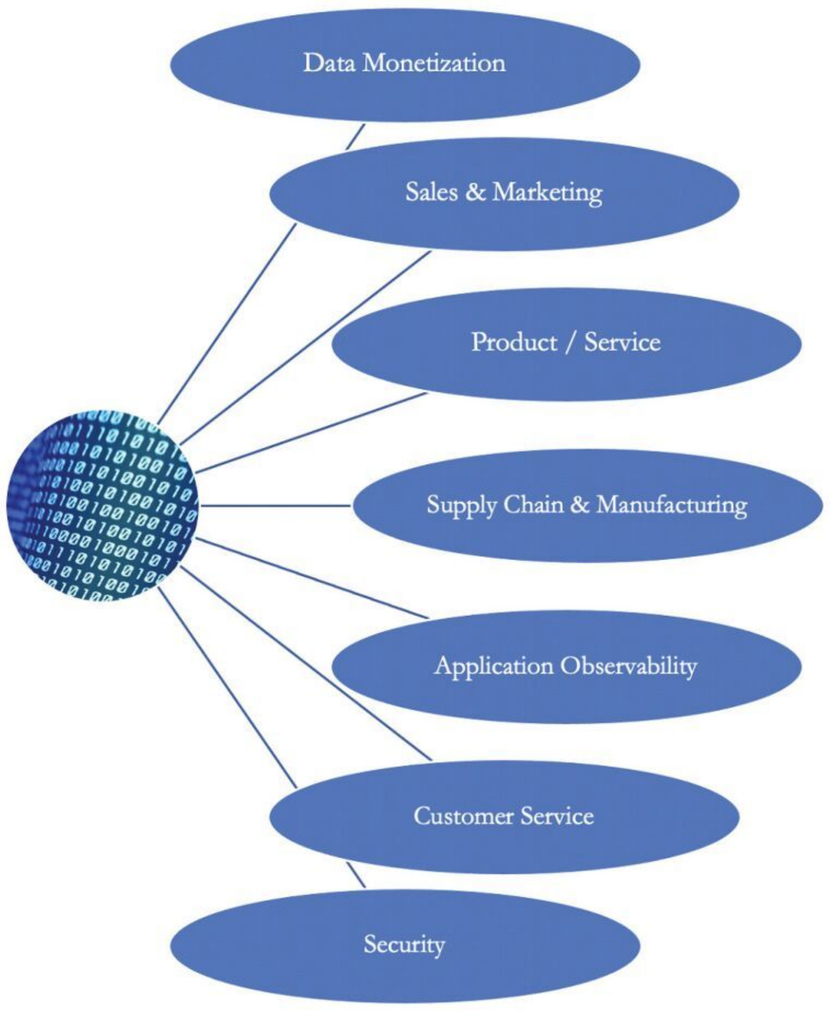
According to Wikipedia, "Data monetization, a form of monetization, may refer to the act of generating measurable economic benefits from available data sources." Let us briefly look at the open source tools that can help with this.

Digital transformation has led to organisations of all sizes getting too much data! This data is generated from mobiles, the web, middleware, business applications, cloud services, SaaS, backend security, monitoring, and much more, and is growing exponentially. Data stakeholders struggle to make meaning out of data lakes, ever-growing data warehouses, Big Data, etc. They develop data management strategies only to realise they have not been able to cover a good amount of new data that was generated by the time their program went live. So let us look at how to develop data monetization strategies across the organisation.

Every business's success depends on data more than ever. However, organisations are struggling to find a workable business strategy around data. The key to successful innovation through data relies upon finding insights. Intersections of data streams, typically, are scattered, making it harder to do this. Also, in the siloed world, data is centred around its primary use application. So there is a need for flexible tools that let data reside where it is in its original format, adhering to local constraints of integrity.



Image Source: https://www.freepik.com



Open source data frameworks and platforms

There are various open source tools that support different needs of data, including data monetization, in any given organisation, some of which are listed below.

Data lake

- The open source storage framework Delta Lake supports the construction of Lakehouse architectures using APIs for Scala, Java, Rust, Ruby, and Python as well as compute engines like Spark, PrestoDB, Flink, Trino, and Hive.
- Kylo is an enterprise-ready data lake management platform for self-service data ingesting and data preparation with integrated metadata management, governance, and security. It is inspired by Think Big's 150+ Big Data implementation projects.

Data warehouse

- Hydra is a Postgres-based open source data warehouse with the motto 'Data-driven decisions at every level'.
- Greenplum Database is an advanced, fully functional, open source data warehouse. It offers strong and quick analytics. A sophisticated cost based query optimiser powers this database, which is specifically designed for Big Data analytics on massive data volumes.

Data mart

- PostgreSQL, commonly referred to as Postgres, is a relational database management system that is open source, and places a strong emphasis on flexibility and SQL compliance.
- One of the most well-known open source relational databases is

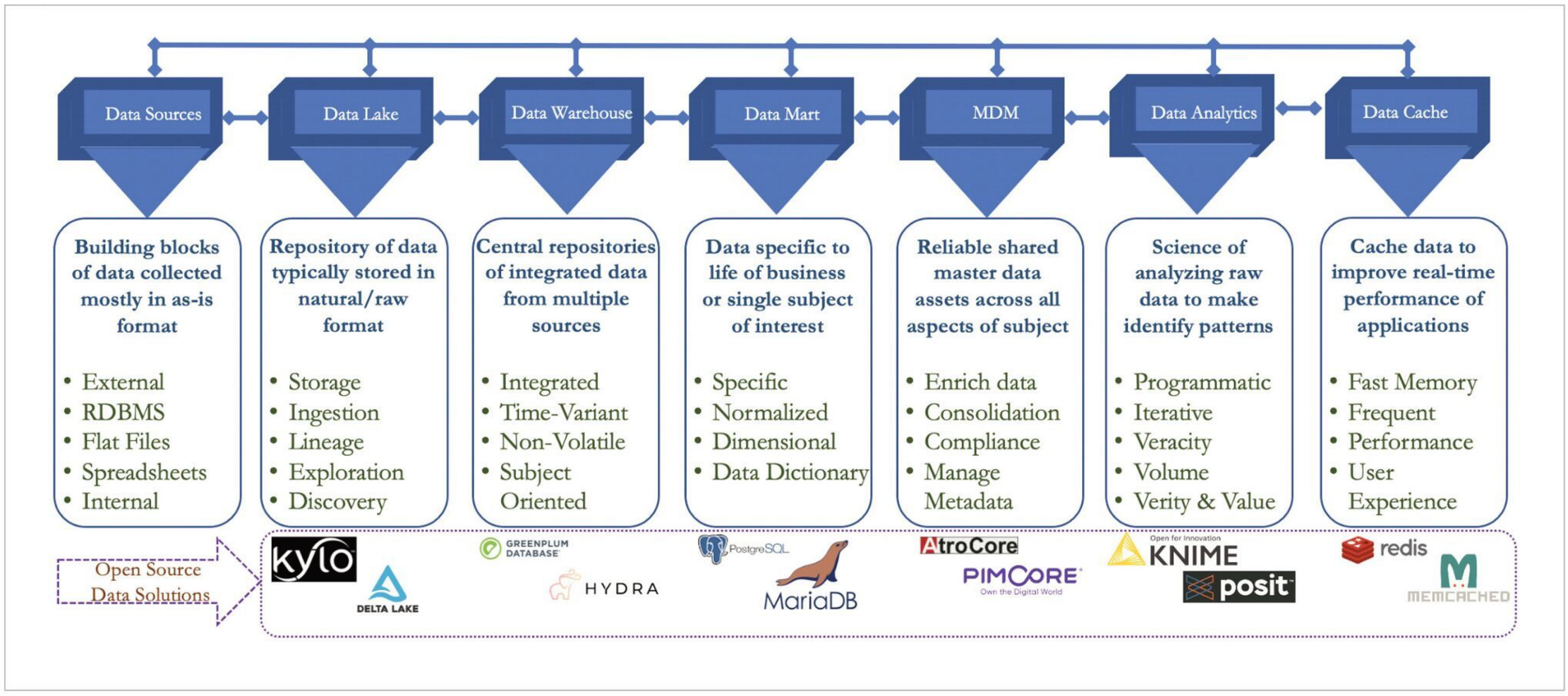


Figure 1: Enterprise data strategy powered by open source

MariaDB Server. Performance, stability, and openness are its guiding principles, and the MariaDB Foundation guarantees that contributions will be accepted on the basis of their technical quality.

Master Data Management (MDM)

- AtroCore is free open source MDM software offered under the GPLv3 license. It is a software ecosystem created for the quick development of ERP-like and responsive web-based business apps. It is a superb tool for quick and cost-effective application development because of its configurable options and strong out-of-the-box capability.
- Pimcore enables the storing of master data, clients, goods, and more in one place. The Pimcore MDM tool has the distinctive quality of not being appropriate for small-sized businesses. It promises continuous data quality improvement and a data quality strategy.

Data analytics

- KNIME Analytics Platform is a free and open source program used for data science. It makes building data science workflows and reusable components accessible to everyone.
- RStudio is now Posit, and its goal remains the same. It wants to improve data science by making it more transparent, logical, approachable, and collaborative.

Data cache

- Memcached is a distributed memory object caching technology with great performance that was initially created to speed up dynamic web applications by reducing database load. It gives you the ability to transfer memory from parts of your system where you have more memory than you require, to sections where you have less memory than you require.
- Redis is a distributed, in-memory, key-value database, cache, and message broker with optional durability that is used as an in-memory data structure store. Various abstract data structures, including strings, lists, maps, sets, sorted sets, HyperLogLogs, bitmaps, streams, and spatial indices, are supported by Redis.

In a world that's increasingly data-rich, we need to enable the evolution of technology and culture in parallel, to discover new insights that guide business strategy, ignite innovation, and redefine how organisations achieve success. END

References

- *https://redis.com/*
- *https://memcached.org/*
- *https://posit.co/*
- *https://www.knime.com/*
- *https://www.plytix.com/*
- *https://atrocore.com/*
- *https://mariadb.org/*
- *https://www.postgresql.org/*
- *https://greenplum.org/*
- *https://hydras.io/*
- *https://kylo.io/*
- *https://delta.io/*

By: Bala Kalavala

The author is a seasoned emerging technical architect, evangelist, thought leader, and sought-after keynote speaker. He currently works as a distinguished member of the technical staff and head of Enterprise Architecture practice as chief architect in a global technology consulting firm.

Installing and Setting Up UrBackup on Ubuntu

This is the age of data. Keeping a backup of this data is critical for every business. This short tutorial explains how to install and use UrBackup, a simple open source tool, for data backup.



Today, data is the most precious thing any business can have. As a result, data security, data backup, etc, have become equally important. Data backup is basically the process of storing multiple copies of data at different locations, etc. Data recovery means bringing back data from elsewhere to the original location.

UrBackup is a tool that helps us maintain a backup of our data and retrieve it when required. It is a very easy to use open source backup system. Let's learn how to install and use it on our Ubuntu Machine!

We first need to install the required packages as shown in Figure 1, using the command given below:

```
sudo apt install gnupg2 curl software-properties-common
```

Now we can add the *repo* of the UrBackup application using the following command, as shown in Figure 2:

```
sudo add-apt-repository ppa:uroni/backup
```

```
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ sudo apt install gnupg2 curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.96.24.32.18).
software-properties-common set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-hwe-5.4-headers-5.4.0-100 linux-hwe-5.4-headers-5.4.0-104 linux-hwe-5.4-headers-5.4.0-105 linux-hwe-5.4-headers-5.4.0-107 linux-hwe-5.4-headers-5.4.0-109 linux-hwe-5.4-headers-5.4.0-117
  linux-hwe-5.4-headers-5.4.0-121 linux-hwe-5.4-headers-5.4.0-122 linux-hwe-5.4-headers-5.4.0-124 linux-hwe-5.4-headers-5.4.0-125 linux-hwe-5.4-headers-5.4.0-42 linux-hwe-5.4-headers-5.4.0-99
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  curl gnupg2
0 upgraded, 2 newly installed, 0 to remove and 30 not upgraded.
Need to get 164 kB of archives.
After this operation, 450 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 curl amd64 7.58.0-2ubuntu3.21 [159 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 gnupg2 all 2.2.4-1ubuntu1.6 [5,300 B]
Fetched 164 kB in 1s (112 kB/s)
Selecting previously unselected package curl.
(Reading database ... 506107 files and directories currently installed.)
Preparing to unpack .../curl_7.58.0-2ubuntu3.21_amd64.deb ...
Unpacking curl (7.58.0-2ubuntu3.21) ...
Selecting previously unselected package gnupg2.
Preparing to unpack .../gnupg2_2.2.4-1ubuntu1.6_all.deb ...
Unpacking gnupg2 (2.2.4-1ubuntu1.6) ...
Setting up curl (7.58.0-2ubuntu3.21) ...
Setting up gnupg2 (2.2.4-1ubuntu1.6) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$
```

Figure 1: Install packages

```
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ sudo add-apt-repository ppa:uroni/urbackup
 Efficient Client-Server Backup system for Linux and Windows. A client for Windows lets you backup open files and complete partition imag
on either Windows or Linux servers.
 More info: https://launchpad.net/~uroni/+archive/ubuntu/urbackup
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:2 http://packages.microsoft.com/repos/code stable InRelease
Hit:3 https://packages.microsoft.com/repos/ms-teams stable InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:7 http://ppa.launchpad.net/obsproject/obs-studio/ubuntu bionic InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:9 http://ppa.launchpad.net/strukturag/libheif/ubuntu bionic InRelease
Get:10 http://ppa.launchpad.net/uroni/urbackup/ubuntu bionic InRelease [15.4 kB]
Get:11 http://ppa.launchpad.net/uroni/urbackup/ubuntu bionic/main i386 Packages [572 B]
Get:12 http://ppa.launchpad.net/uroni/urbackup/ubuntu bionic/main amd64 Packages [576 B]
Get:13 http://ppa.launchpad.net/uroni/urbackup/ubuntu bionic/main Translation-en [252 B]
Fetched 16.8 kB in 3s (5,668 B/s)
Reading package lists... Done
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$
```

Figure 2: Add the *repo* of UrBackup

```
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ sudo apt update
Hit:1 http://packages.microsoft.com/repos/code stable InRelease
Hit:2 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 https://packages.microsoft.com/repos/ms-teams stable InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:6 http://ppa.launchpad.net/obsproject/obs-studio/ubuntu bionic InRelease
Hit:7 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:8 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:9 http://ppa.launchpad.net/strukturag/libheif/ubuntu bionic InRelease
Hit:10 http://ppa.launchpad.net/uroni/urbackup/ubuntu bionic InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
30 packages can be upgraded. Run 'apt list --upgradable' to see them.
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$
```

Figure 3: Updating *apt*

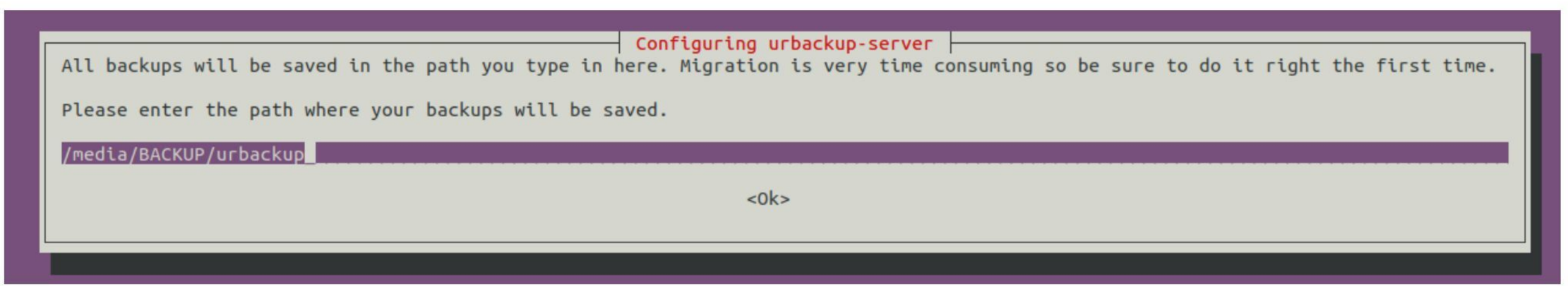


Figure 4: Backup location

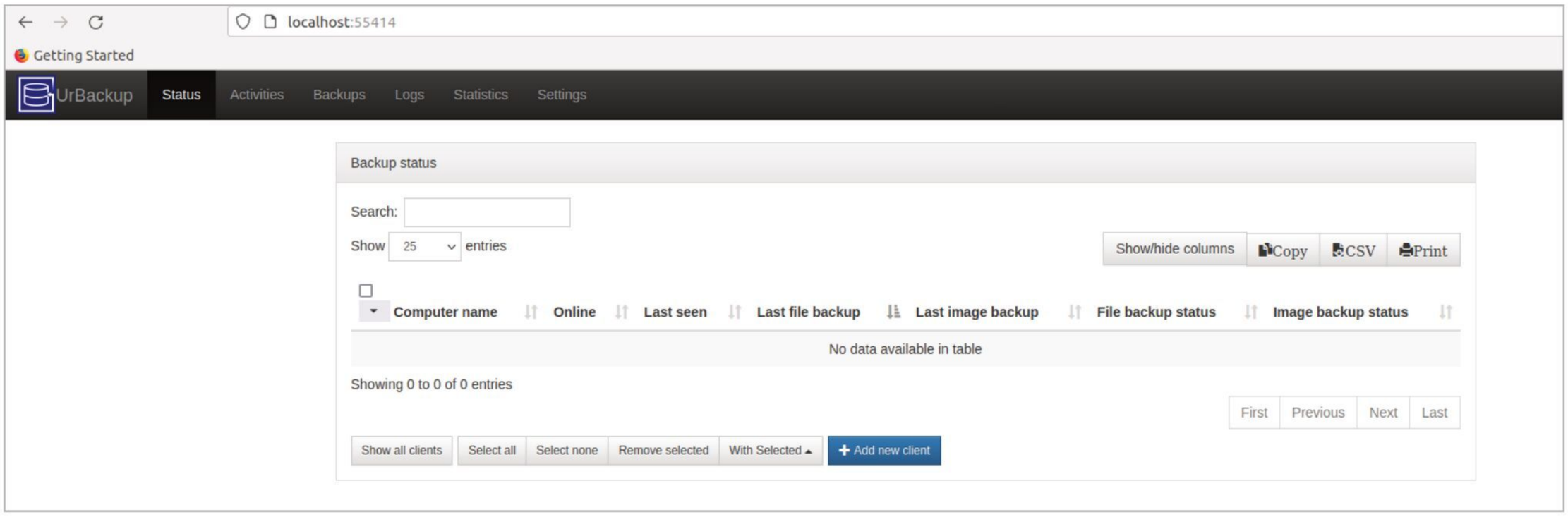


Figure 5: UrBackup on the browser

Once that is done, update our *apt* as shown in Figure 3, using the following command:

```
sudo apt update
```

Now go ahead and install the UrBackup server using the command given below:

```
sudo apt install urbackup-server
```

You will be asked the location where your backup needs to be stored; enter the location you desire, as shown in Figure 4.

Now that the installation and configuration are done, you can open UrBackup, set up the different options and start backing data up, as shown in Figure 5, by going to the port *http://localhost:55414/* on your local host on your browser.

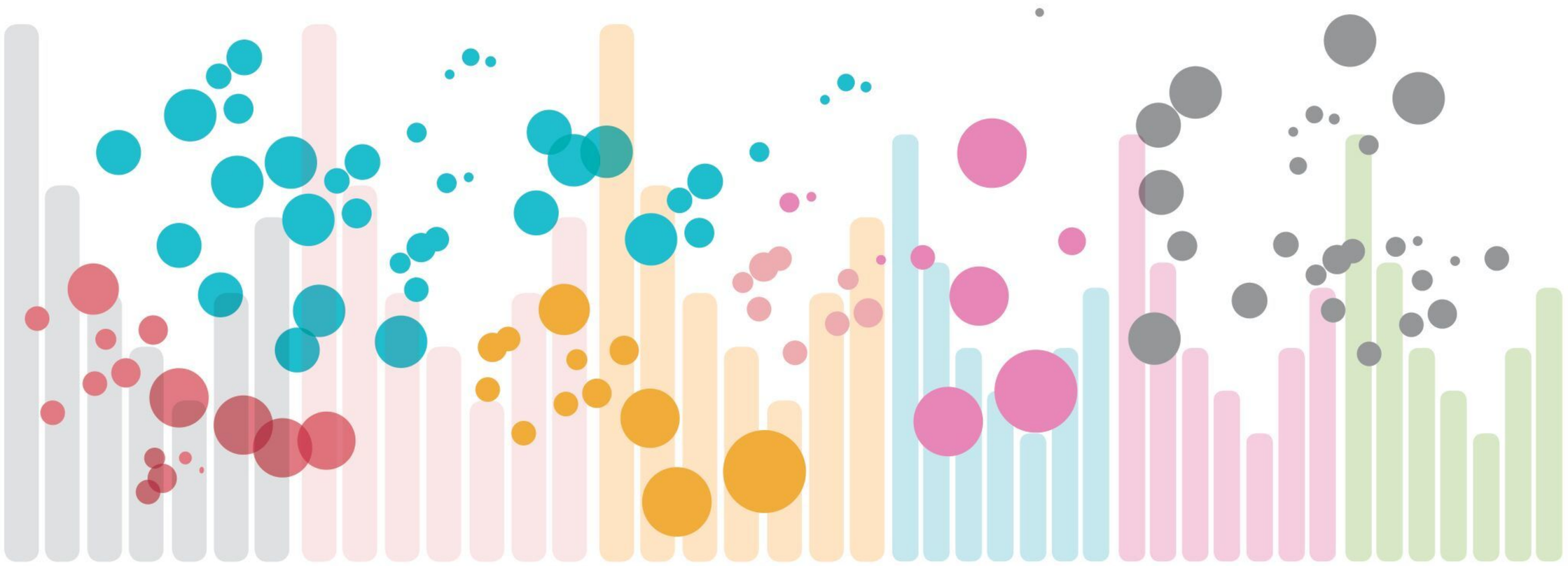
You can explore the rest of the options in the web interface to set up an effective backup of your data. The system can also be configured along with a NAS drive to create a complete backup system of a small office/home office with multiple endpoints. END

By: Jishnu Saurav Mittapalli

The author is a full stack Web developer, interested in modern technologies like artificial intelligence and machine learning. He is currently working in the field of natural language processing.

RAWGraphs: The No-Code Data Visualisation Tool

We need specific and effective tools to make sense of the vast amounts of data being generated today. This article looks at an interesting open source data visualisation tool called RAWGraphs, which helps build professional quality visualisation without requiring any code. Let's see how.



The volume and variety of data being generated each day is increasing exponentially. The success or failure of a software application depends on the efficiency with which it handles data. A major challenge this massive, heterogeneous data poses is the difficulty in inferring patterns. Data visualisation renders the data in a visual form, which is easier for us to understand. As the human brain is better at processing visual information in comparison to textual and numeric data, with data visualisation it becomes easier to understand the data.

Data visualisation today has applications across various domains such as healthcare, finance, tourism, banking, cybersecurity and e-commerce. In fact, it is difficult to think of a domain that is not benefited by data visualisation. There are many tools that help to carry out data visualisation. Some of them are listed below:

- RAWGraphs
- D3.js
- RapidMiner
- Chart.js
- Google Charts

Each of these tools has a specific purpose and features. This article focuses on an interesting data visualisation tool called RAWGraphs. This free and open source tool has been developed jointly by Density Design, Calibro and Inmagik. The project is led and maintained by the Density Design Research Lab at Politecnico di Milano. Released to the public in 2013, it has evolved into an important tool in the domain of data visualisation. Calibro and Inmagik joined the team in 2019.

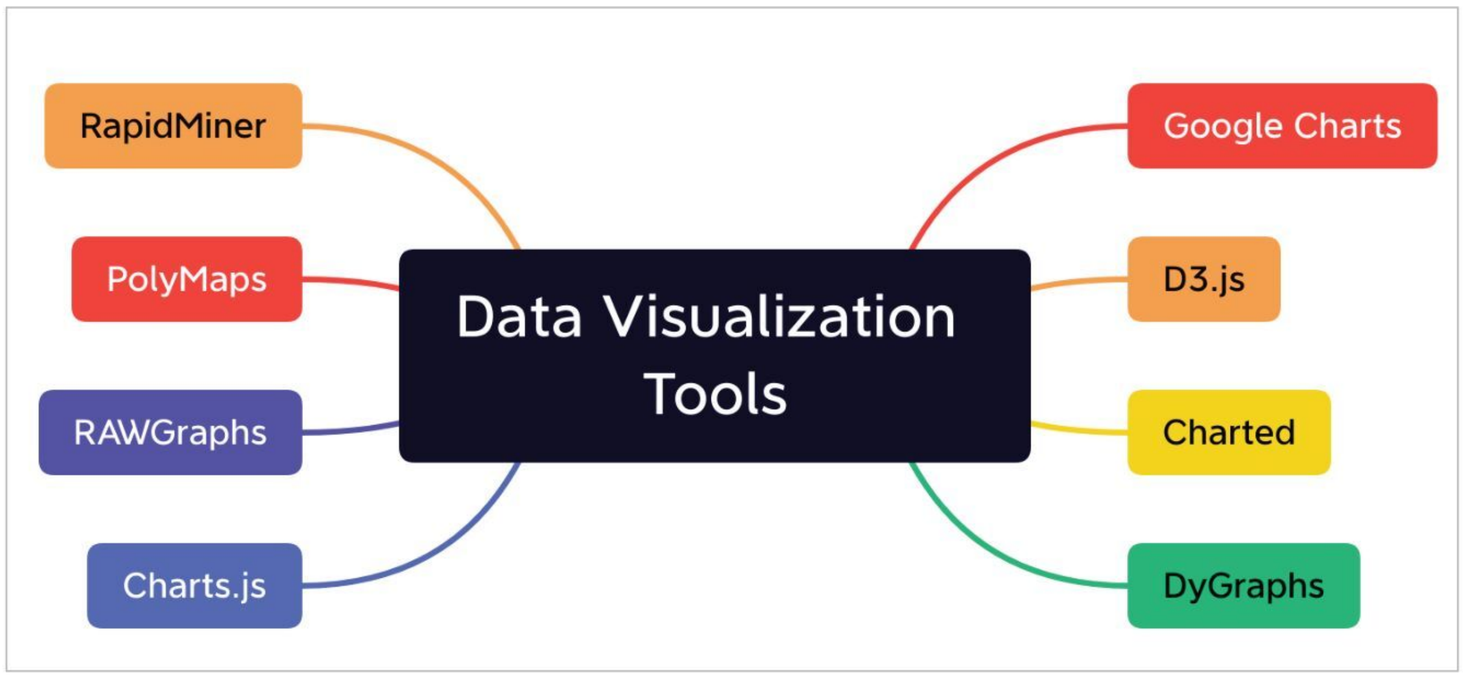


Figure 1: Data visualisation tools

That this is a proven data visualisation tool is evident from the fact that it was presented as a research paper in one of the leading conferences in this domain (*https://doi.org/10.1145/3125571.3125585)*. RAWGraphs is supported by both academia and industry.

The primary objective of RAWGraphs is to make the complex data visualisation process easy for everyone. This is achieved by enabling visualisation without requiring any code. RAWGraphs is designed to bridge the gap between spreadsheet applications and vector graphics editors. Anyone with a basic knowledge of computing can use it easily.

Key features

Open source: RAWGraphs is free and open source. The source code repository can be accessed at *https://github.com/rawgraphs/rawgraphs-core.*

Visualisation options: Depending on the nature of the data and the domain, one can choose from different visualisation options, making RAWGraphs a very useful tool.

Data safety: The official documentation states that the data processing happens only at the client browser and that data is secure.

Ability to export in various formats: RAWGraphs provides the option to export the visualisation either as a vector or raster image.

No code; no installation: RAWGraphs is available as a web app, which can be accessed directly from your web browser. It doesn't require users to configure anything in their system. This is a major benefit of this tool.

RAWGraphs visualisation pipeline

RAWGraphs has clearly defined the visualisation pipeline. Users have to go through each stage and select the options as per their needs to build effective visualisation. The five different stages in the data visualisation

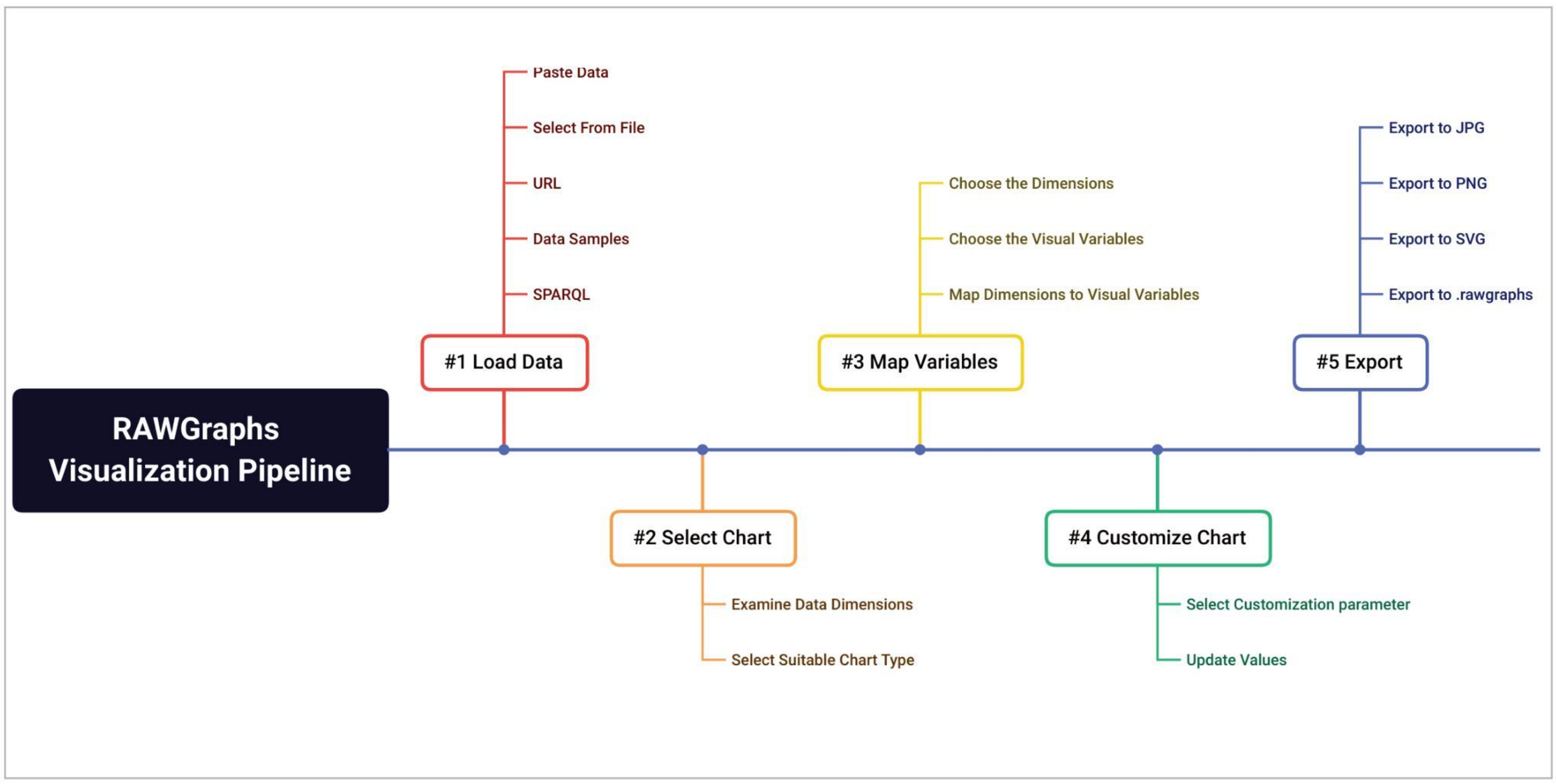


Figure 2: RAWGraphs – visualisation pipeline

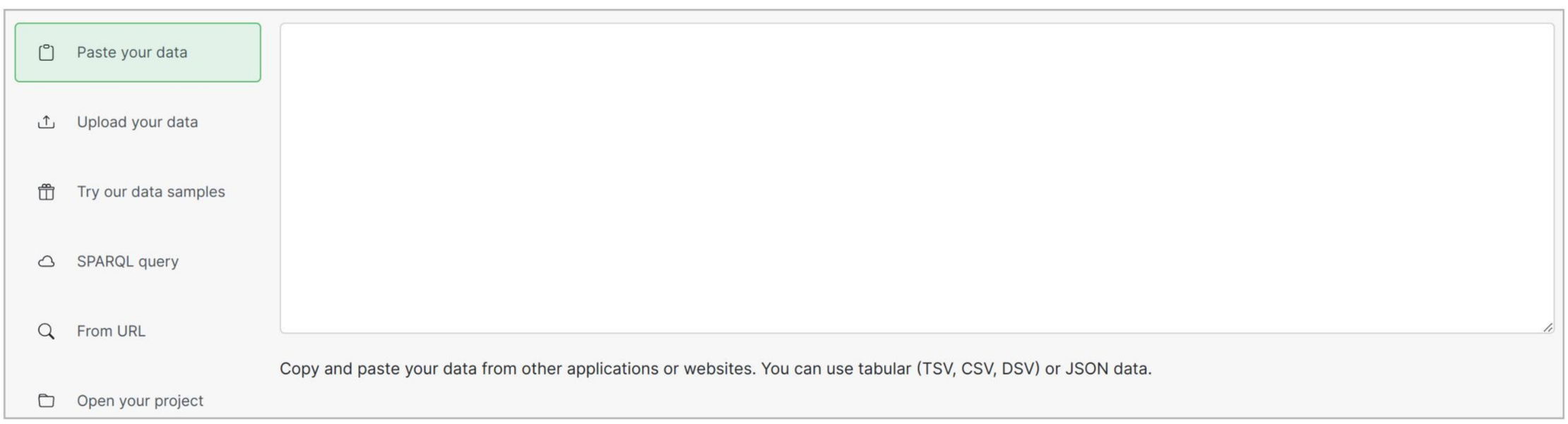


Figure 3: Data loading options

DATA PARSING OPTIONS

Column separator: \t Tab

Thousands separator: ,

Decimals separator: .

Date Locale: en-GB

DATA TRANSFORMATION

Stack on: Column

Reset

	Letter	Language	Frequency (%)
105	a	Esperanto	12.12
106	i	Esperanto	10.01
107	e	Esperanto	9
108	o	Esperanto	8.78
109	n	Esperanto	7.96
110	l	Esperanto	6.1
111	s	Esperanto	6.09

Figure 4: Sample data from a repository

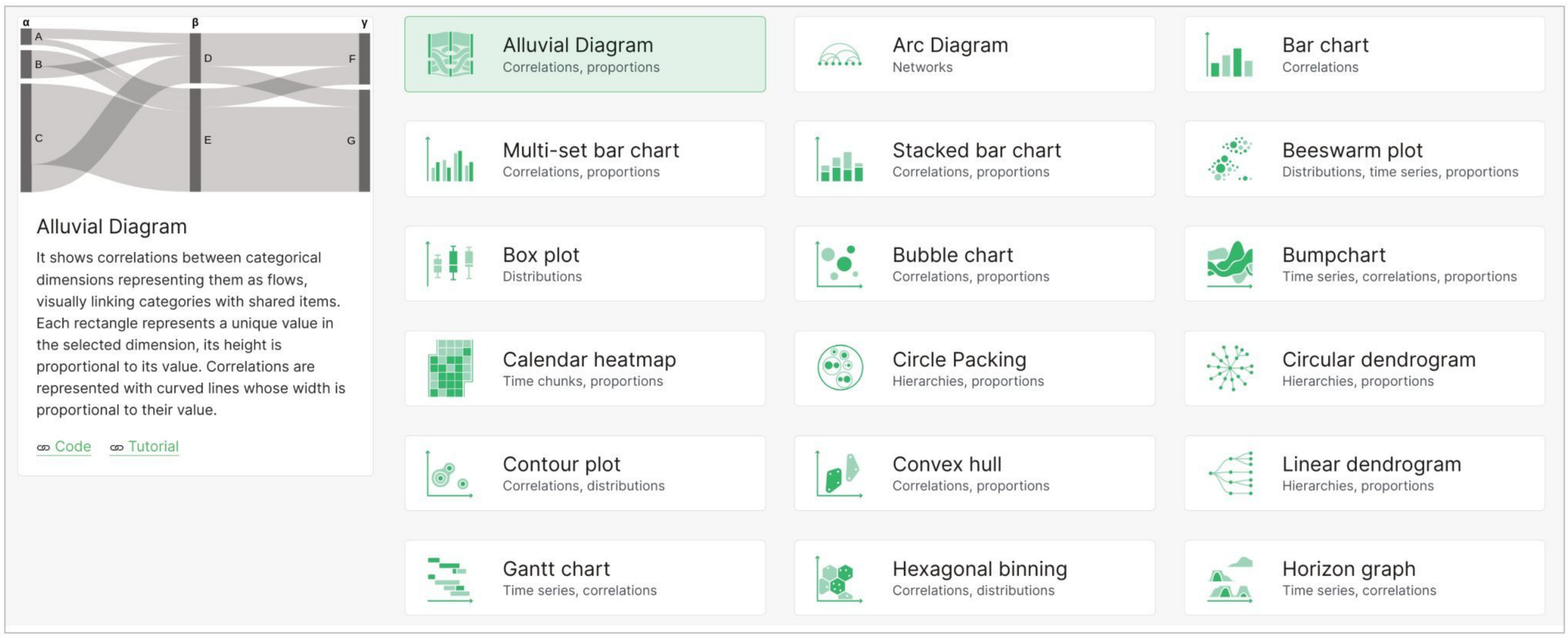


Figure 5: Chart types in RAWGraphs

pipeline (Figure 2) of RAWGraphs are:

- Data loading
- Chart selection
- Data dimension mapping to the visual variables of the chart
- Customisation
- Exporting

Data loading

The RAWGraphs web app can be accessed at *https://app.rawgraphs.io/*. The first step in the visualisation pipeline is to load the data. The following options are provided in this step:

- Paste your data
- Upload your data
- Try data samples
- SPARQL query
- From URL
- Open your project

When you select the option 'Upload your data', RAWGraphs allows you to choose from the following file formats:

- *.tsv* (Tab separated values)
- *.csv* (Comma separated values)
- *.dsv* (Delimiter separated values)
- *.json* (JavaScript object notation)

In the case of *.json* files, the level of the nested structure can be selected to transform the data into a table.

Figure 4 shows sample data loaded from the repository 'Letter frequency by language'. This repository has 390 rows (1170 cells).

Chart selection

The second step is to select the chart type. The power of RAWGraphs lies in the variety and specific types of charts it provides. Figure 5 illustrates a set of charts this tool can render.

The best part is that, for each chart type, a clear description along with a demo video is provided, helping users make an informed choice.

The most important task in the data visualisation process is the selection of suitable types of visualisation. Incorrect

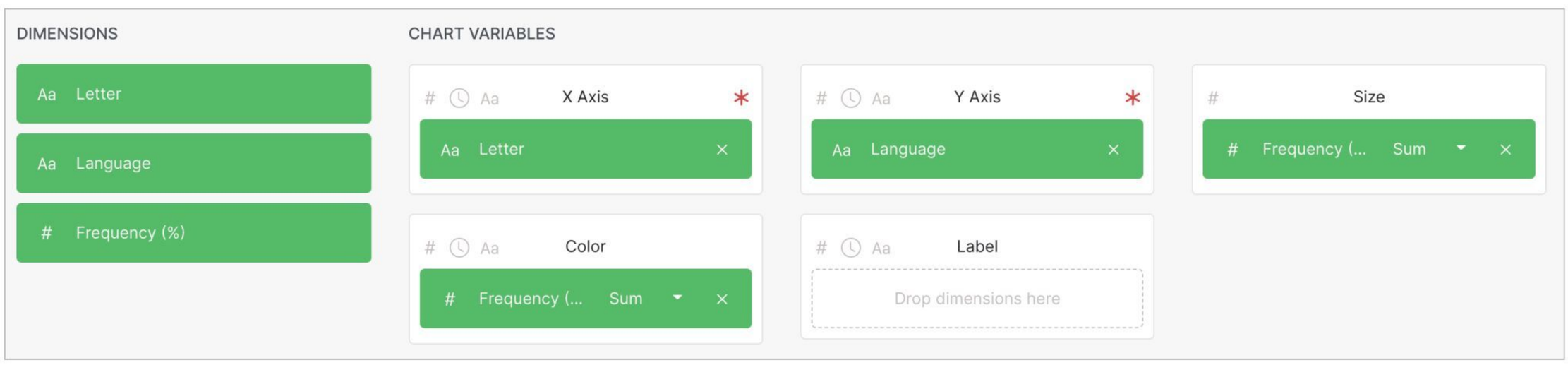


Figure 6: Mapping for 'Matrix Plot'

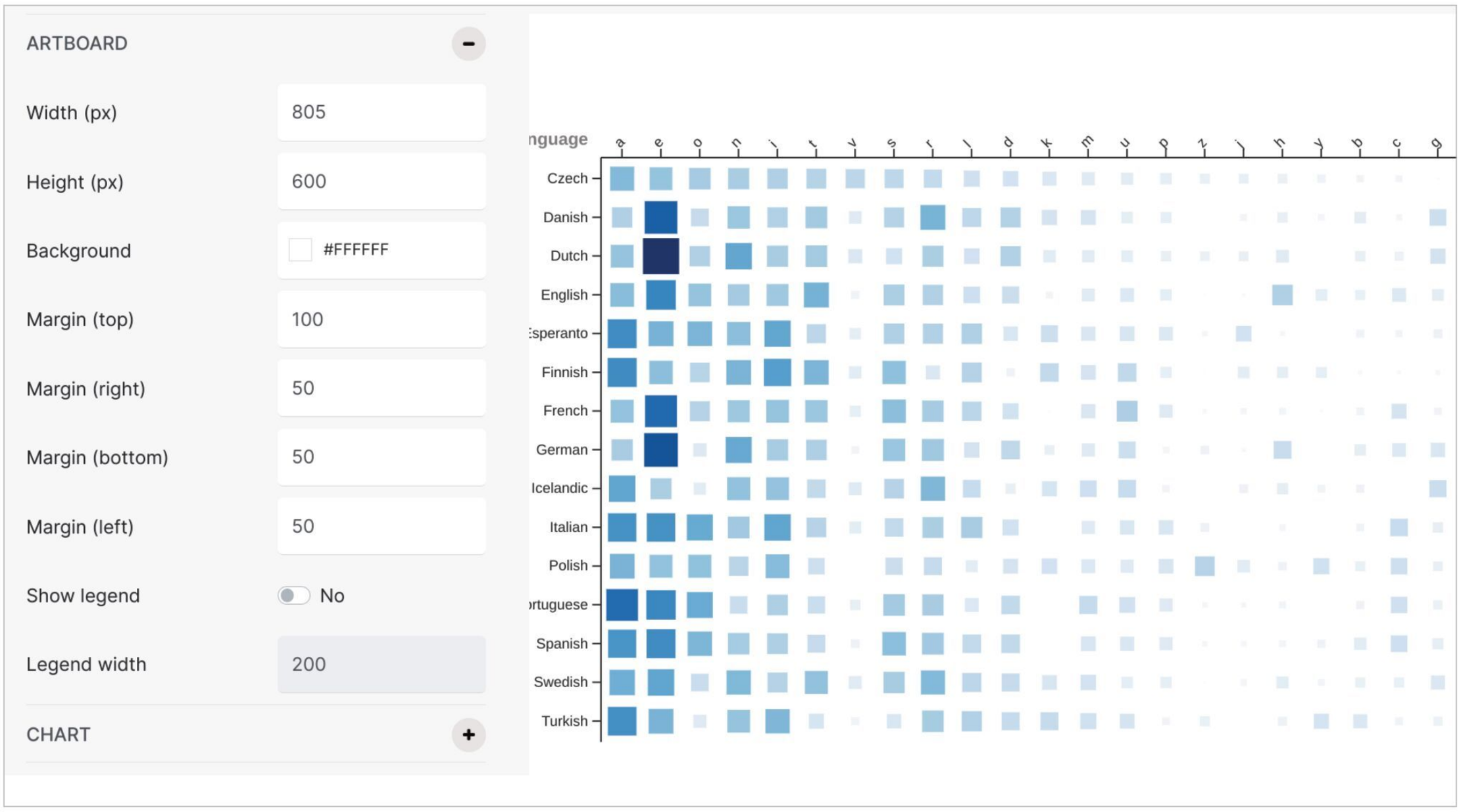


Figure 7: Customisation

selection will lead to suboptimal visualisation, which may not satisfy the goal of comprehending the data. For the data set that we selected (Letter frequency by language), it would be best to select the 'Matrix Plot' for proper visualisation.

Data dimension mapping

The third step is to map the data dimension to the visual variables of the chart. Based on the visualisation that you have selected, RAWGraphs prompts for different dimensions. For the Matrix Plot that we have selected in the previous step, the following are to be mapped:

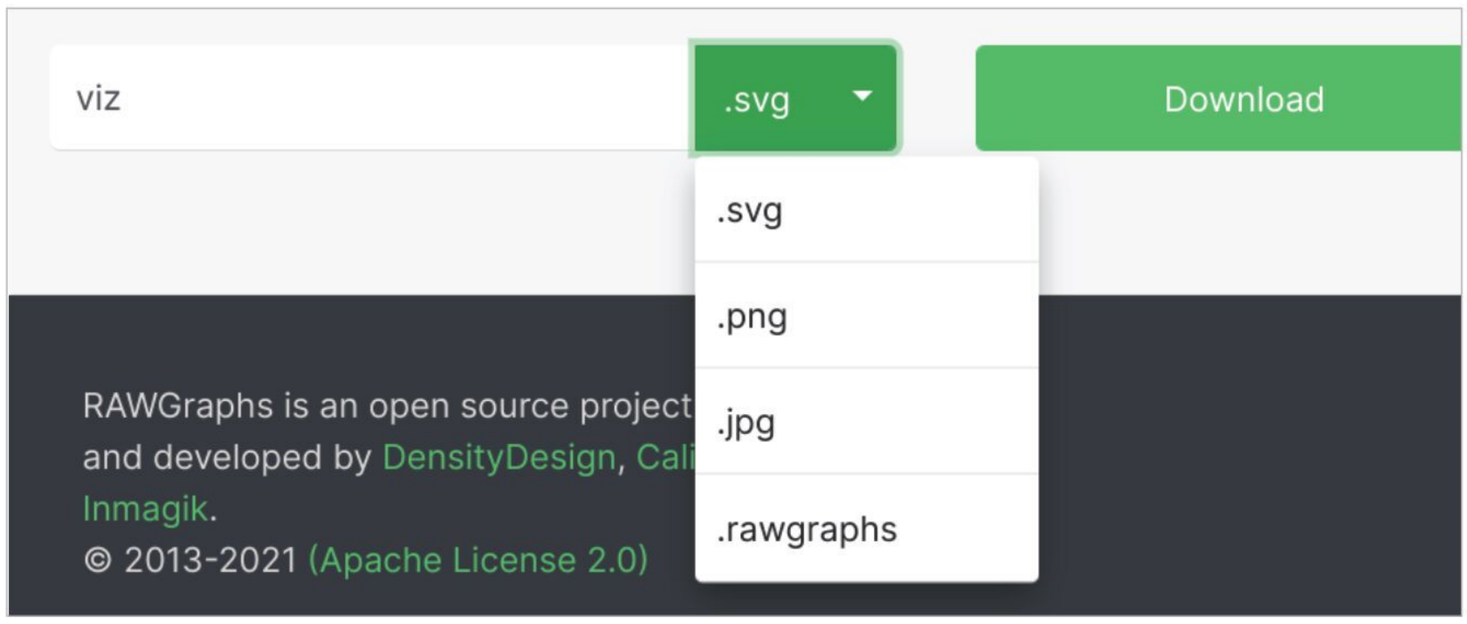


Figure 8: Export visualisation

- X-Axis (Letter)
- Y-Axis (Language)
- Size (Frequency %)
- Colour (Frequency %)

Customisation

This step allows us to change the look and feel of the visualisation. Based on the chart selected, the customisation

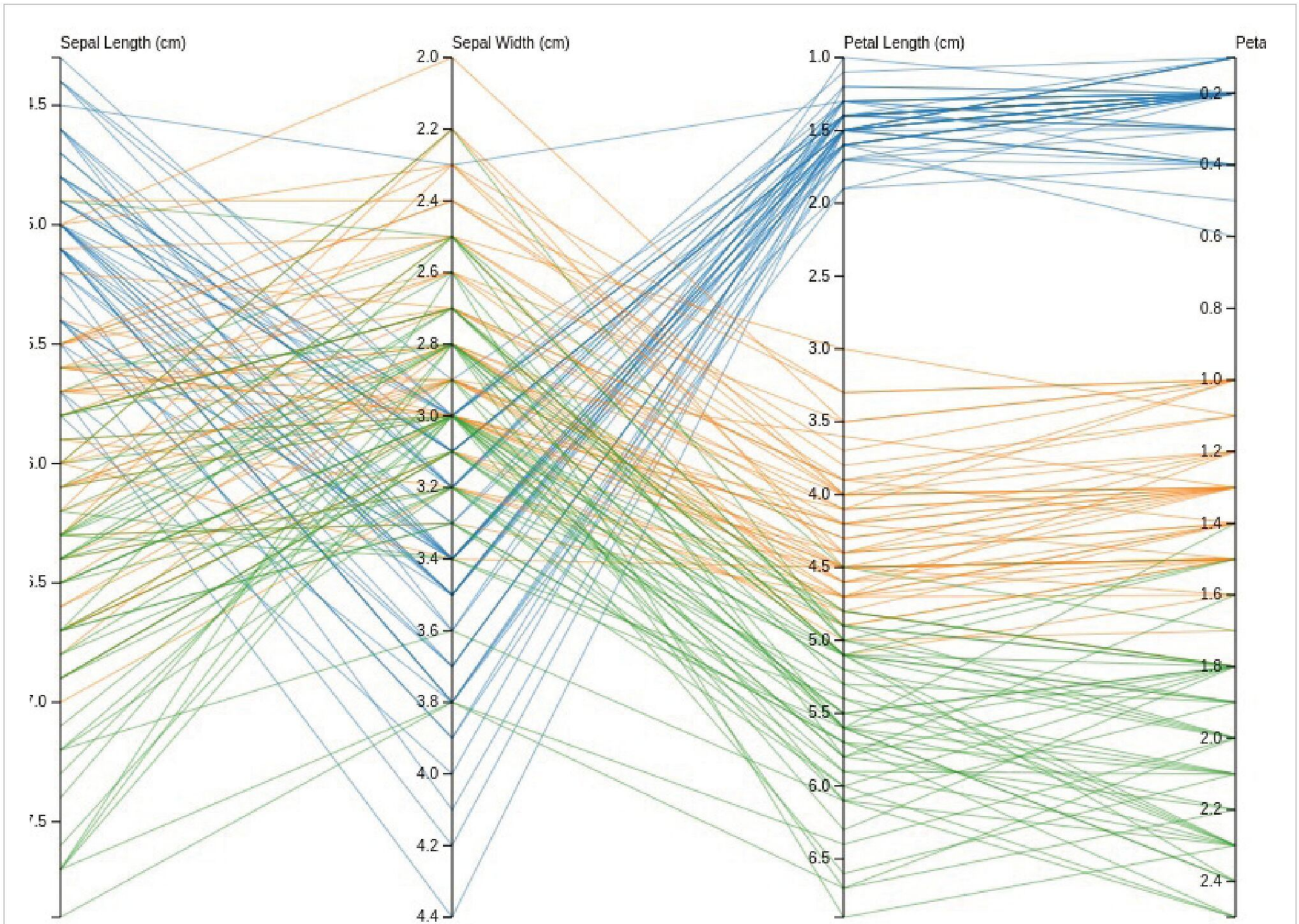


Figure 9: Parallel Coordinates Chart

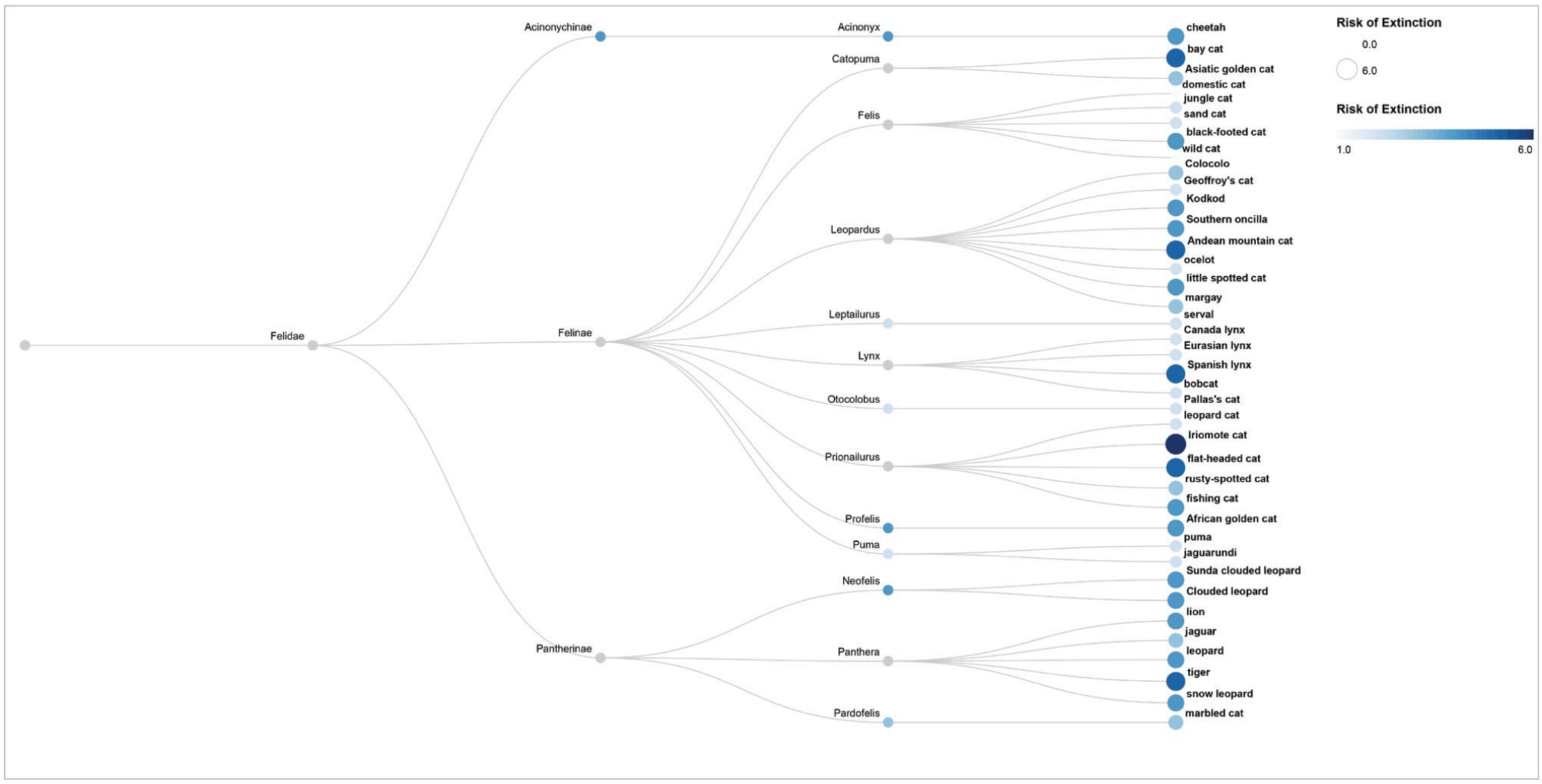
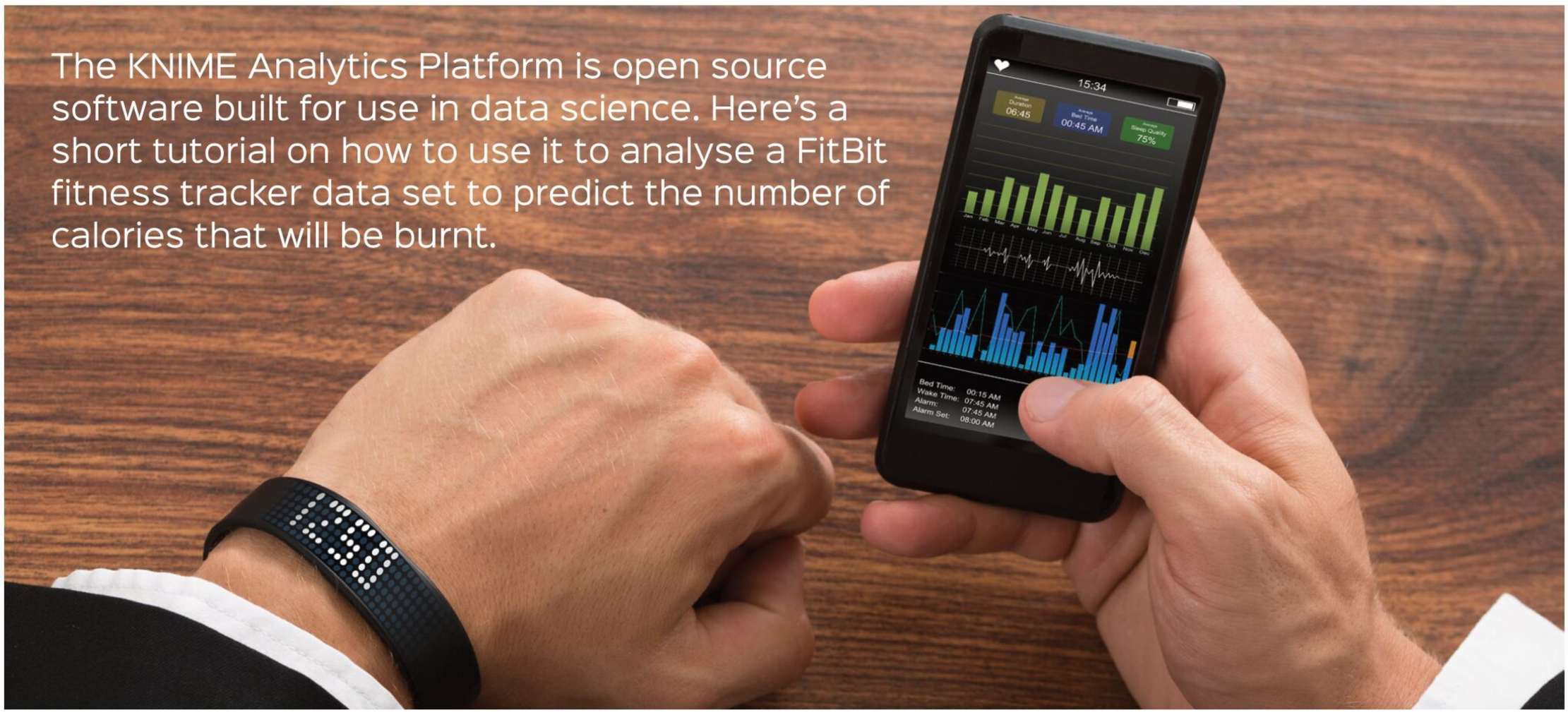


Figure 10: Dendrogram

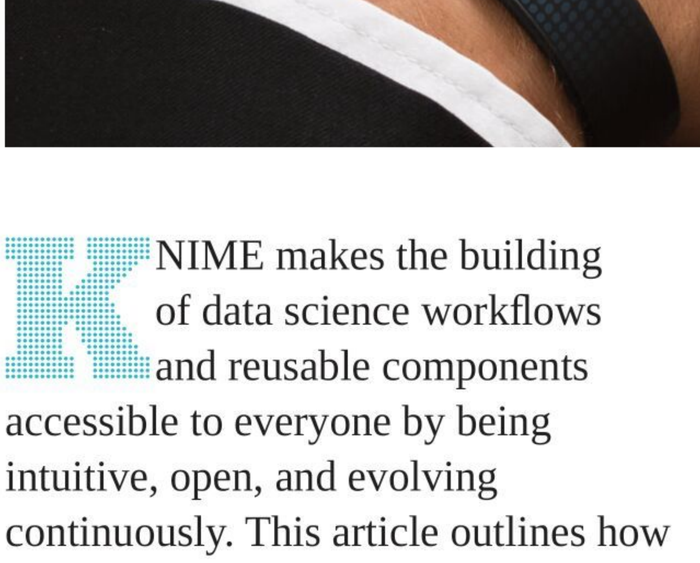
To Continued on page...59

How to Make a Calories Burn Prediction on a Fitness Tracker Data Set



KNIME makes the building of data science workflows and reusable components accessible to everyone by being intuitive, open, and evolving continuously. This article outlines how to make a calorie burn prediction with a random forest predictor through the KNIME Analytics Platform, using a fitness tracker data set.

The KNIME nodes that will be needed for this are listed below:



Building the fitness tracker flow

1. Download the 'FitBit fitness tracker data' data set from *https://www.kaggle.com/arashnic/fitbit*.
2. This data set consists of 15 column attributes, namely: Id, Activity date, Total steps, Total distance, Logged activity distance, Very active distance, Moderately active distance, Light active distance, Sedentary active distance, Very active minutes, Fairly active minutes, Lightly active minutes, Sedentary minutes and Calories. All of the first 14 columns determine the individual's overall calories burnt in a day. With this data set, we can identify if an individual has a high, low, or good calorie burn for the day.
3. Drag and drop the *.csv* file in the KNIME platform to get the CSV Reader node.
4. Connect this with the Column filter and Missing Value node for preprocessing the data. In the Column filter, exclude the ActivityDate column, as the date doesn't matter for this prediction. In the Missing Value node, enter all the missing values as the mean of the entire column.
5. Connect the Missing Value node with the Rule Engine node. Here, give the rules as follows:
   - If Calories > 2500 => High calories burn
   - If Calories < 2000 => Low calories burn
   - If Calories >= 2000 and Calories <= 2500 => Good calories burn.

Append this data into a new column named 'Calories burn'.

6. Connect this with the Box Plot, Linear Correlation and X-Partitioner nodes.
7. In the Box Plot, exclude all the columns except Calories and visualise the plot. We can visualise the robust statistical parameters of minimum, lower quartile, median, upper quartile, and maximum value.
8. The Linear Correlation node calculates a correlation coefficient for each pair of selected columns, i.e., a measure of the correlation of the two variables. Here, all columns are included to calculate the correlation coefficient.
9. The X-Partitioner node is used for performing the cross-validation in the data set. This node is the first in a cross-validation loop. At the end of the loop, there must be an X-Aggregator to collect the results from each iteration. All nodes in between these two nodes are executed as many times as iterations should be performed. We perform 10 validations using random sampling, and the best out of these 10 is given as the output.
10. Connect the first partition with the Random Forest Learner node. Select the Target column as 'Calories burn' and let the rest of the options be the same as the default. Select the split criteria as Information Gain Ratio.
11. Now, connect the Random Forest Predictor node with the second partitioning and the Random Forest Learner node.
12. Connect this node with the X-Aggregator node. This node collects the result from a Predictor node, compares the predicted class and real class, and outputs the predictions for all rows and the iteration statistics. Select the Target column as 'Calories burn' and the Prediction column as 'Prediction (Calories burn)'.
13. Connect this node with the Scorer and Bar Chart nodes.
14. In the Scorer node, select the first column as 'Calories burn' and the second column as 'Prediction (Calories burn)', and save it.
15. In the bar chart, enter the column as 'Prediction (Calories burn)', name the chart, and then save it. Finally, execute all the nodes and view the output.

The complete flow for calorie burn prediction using a random forest predictor is given in Figure 1.

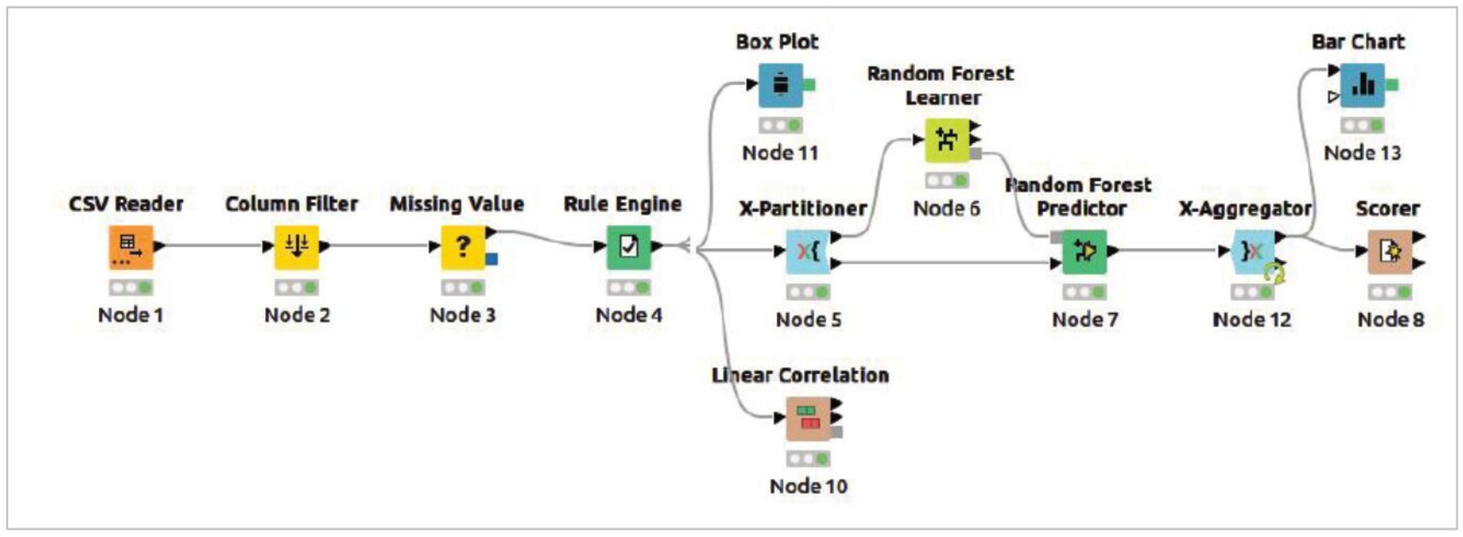


Figure 1: KNIME flow for calorie burn prediction using a random forest predictor

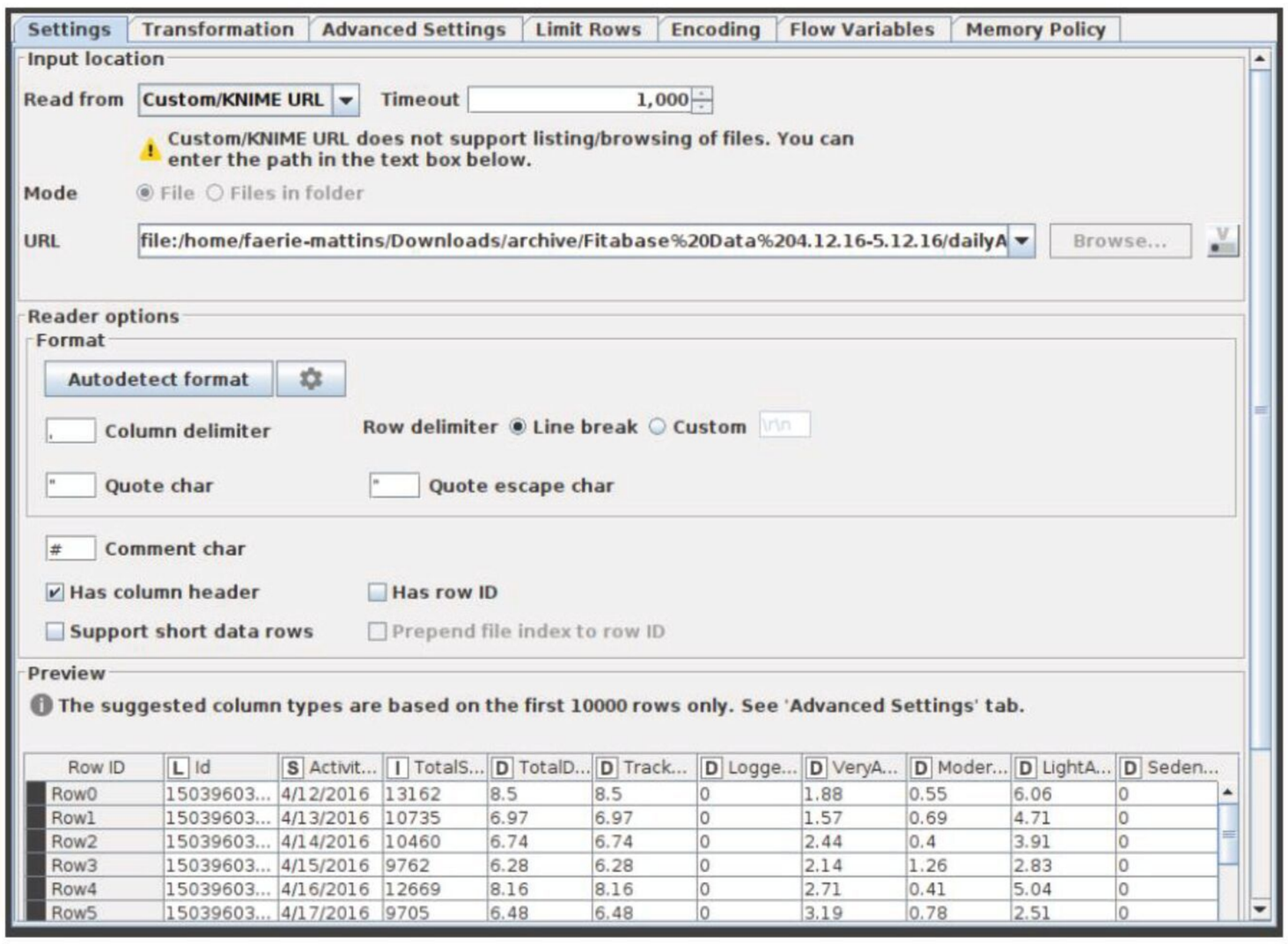


Figure 2: Configuration of CSV Reader node

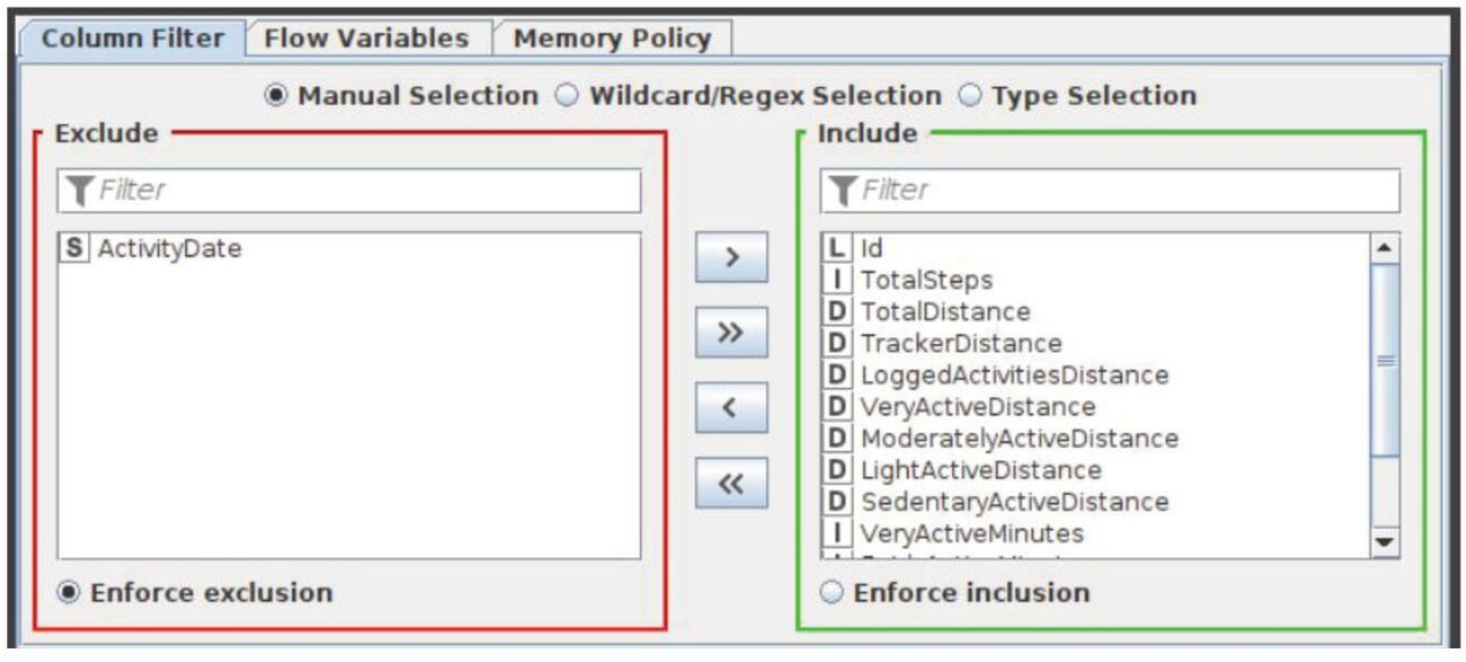


Figure 3: Configuration of Column Filter node

Node configurations

Let us see the details needed for node configuration one by one.

CSV Reader

Figure 2 shows how to configure the CSV Reader.

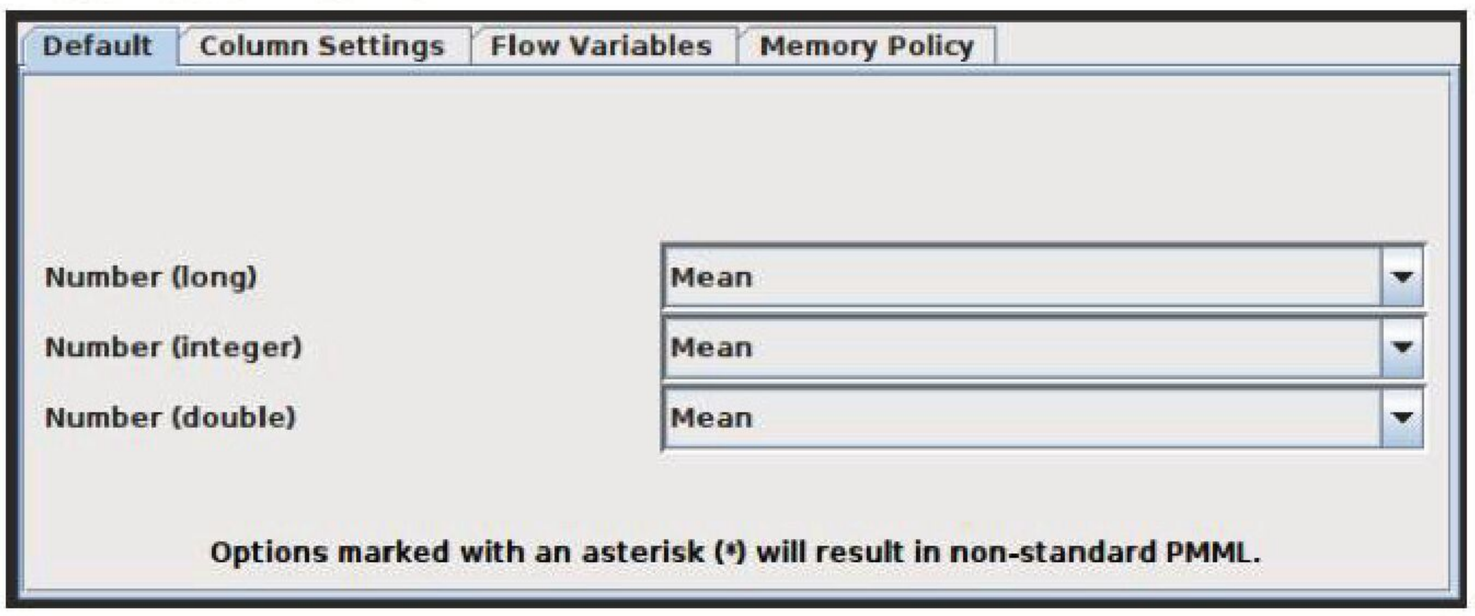


Figure 4: Configuration of Missing Value node

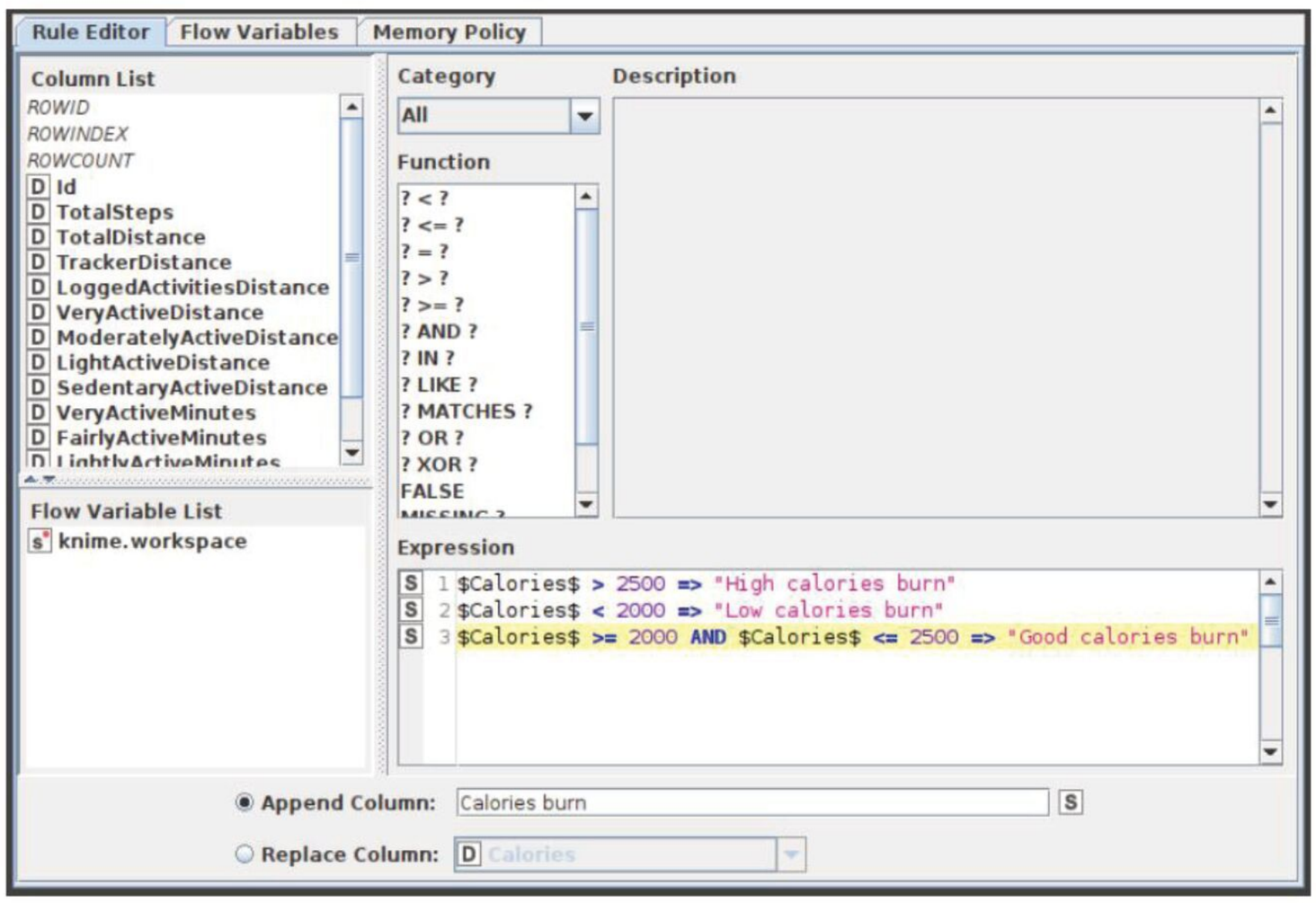


Figure 5: Configuration of Rule Engine node

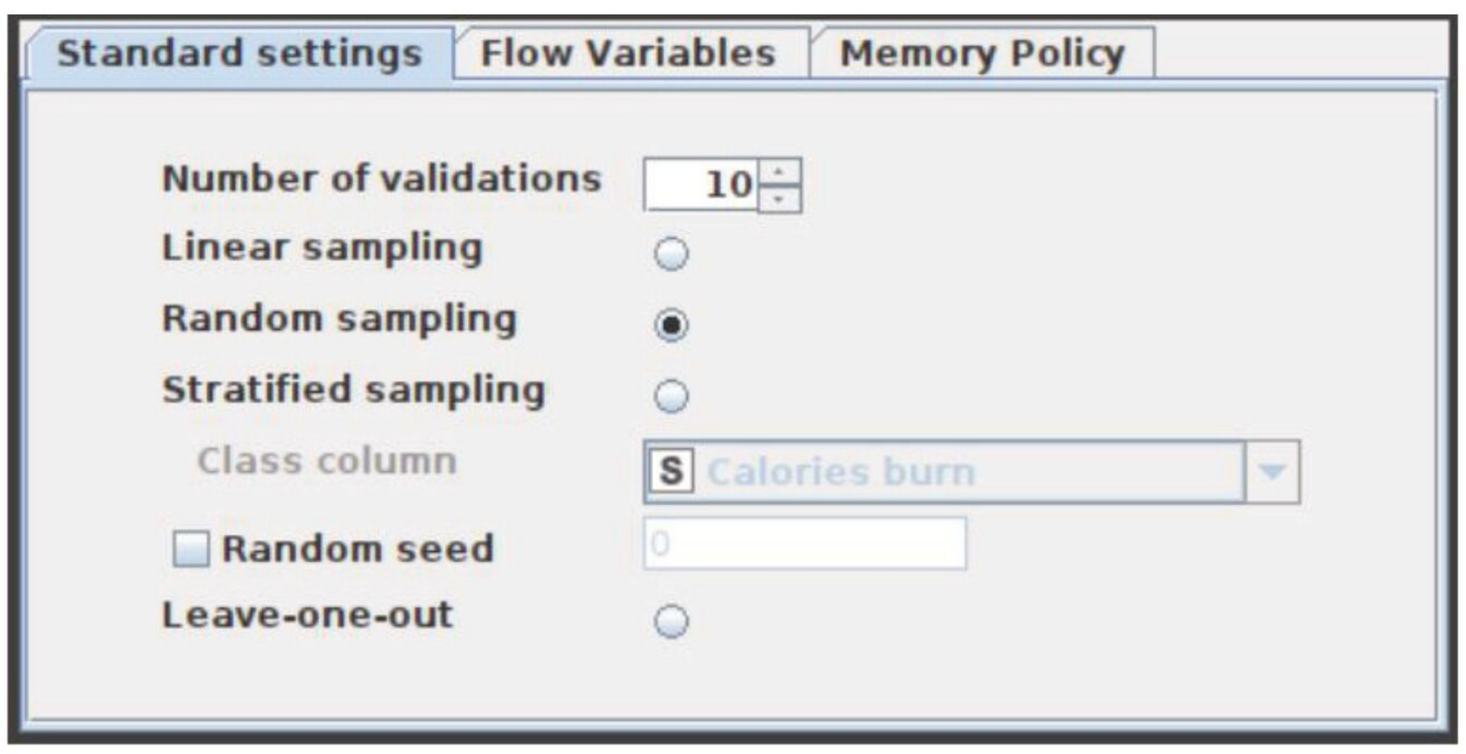


Figure 6: Configuration of X-Partitioner node

Column Filter

This Column Filter node provides three distinct modes for filtering: manually, by name, and by type. Through *Add* and *Remove* buttons, you manually choose which columns to retain and which to exclude. In the name mode, you decide which columns to keep using wild cards and regular expressions. In the type option, you choose which columns to retain based on their type, such as all strings or all integers. The configuration window of this Column Filter node is shown in Figure 3.

Missing Value

This node aids in handling cells in the input table that have missing values. All columns of a particular kind have default handling options available on the dialog's first tab, 'Default'. All columns in the input table that aren't specifically listed in the second tab, 'Individual', have these settings applied to them. Individual settings for each available column are possible using this second tab (thus, overriding the default). Use the second method by selecting the column or group of columns that requires extra treatment, clicking 'Add', and then setting the settings. All covered columns in the column list will be selected when you click on the label with the column name(s). Click the 'Remove' button for this column to get rid of the extra treatment and replace it with the default handling. Asterisk-denoted options (*) will produce non-standard PMML. A warning will be displayed throughout the execution of the node if you choose this option, and the caution label in the dialogue will turn red. Extensions used by non-standard PMML cannot be read by any software other than KNIME. The configuration window of the missing value node is captured and indicated in Figure 4.

Rule Engine

This node attempts to match a list of

user-defined rules to each row in the input table. If a rule is satisfied, the result value is added to a new column. The outcome is determined by the first matching rule in definition order. One line represents each regulation. To insert comments, begin a line with / (comments cannot be placed in the same line as a rule). Anything following / will not be considered a rule. Rules consist of a condition part (antecedent) that must evaluate to *true* or *false* and a consequence (consequent, after the => symbol) that is placed in the new column if the rule matches. A rule's output may be a string (between " and /), a number, a Boolean constant, a reference to another column, or the value of a flow variable. The type column represents the common supertype of all potential outcomes (including the rules that can never match). If no rule is applicable, the result is a missing value. Columns are denoted by their respective names surrounded by $, while numbers are presented in the customary decimal format. Note that strings cannot contain (double) quotation marks. *$$ TypeCharacterAndFlowVarName $$* is used to represent flow variables. The TypeCharacter for double (real) values should be 'D', for integer values 'I', and for strings 'S'. You can manually insert column names and flow variables or use the respective lists in the dialogue. The setup window for the rule engine node is depicted in Figure 5.

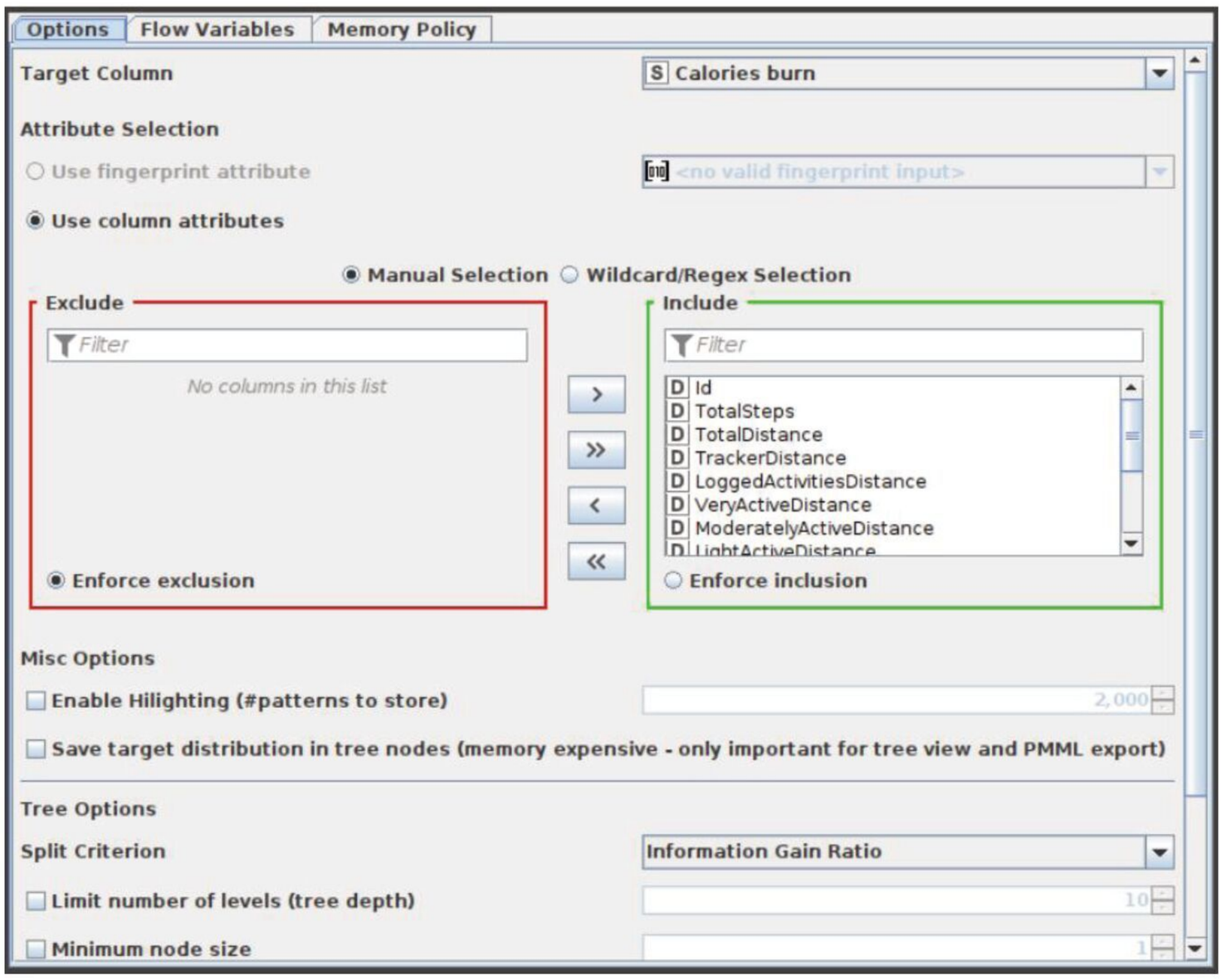


Figure 7: Configuration of Rule Engine node

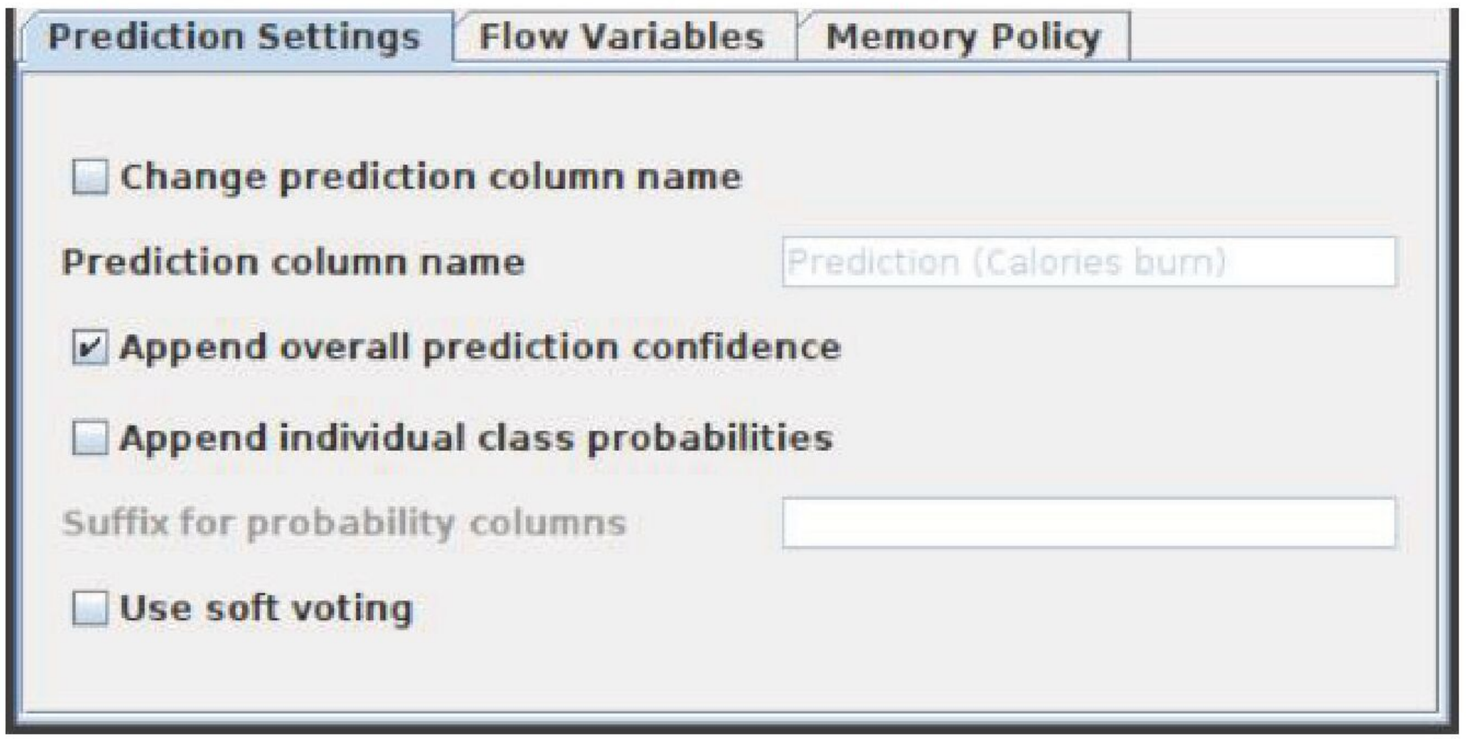


Figure 8: Configuration of Random Forest Predictor node

X-Partitioner

This node begins the cross-validation loop. An X-Aggregator is required at the conclusion of the loop to collect the results of each iteration. All nodes between these two nodes are run as many times as required by the number of iterations. Figure 6 shows the configuration window for the X-Partitioner node that has been selected for this application.

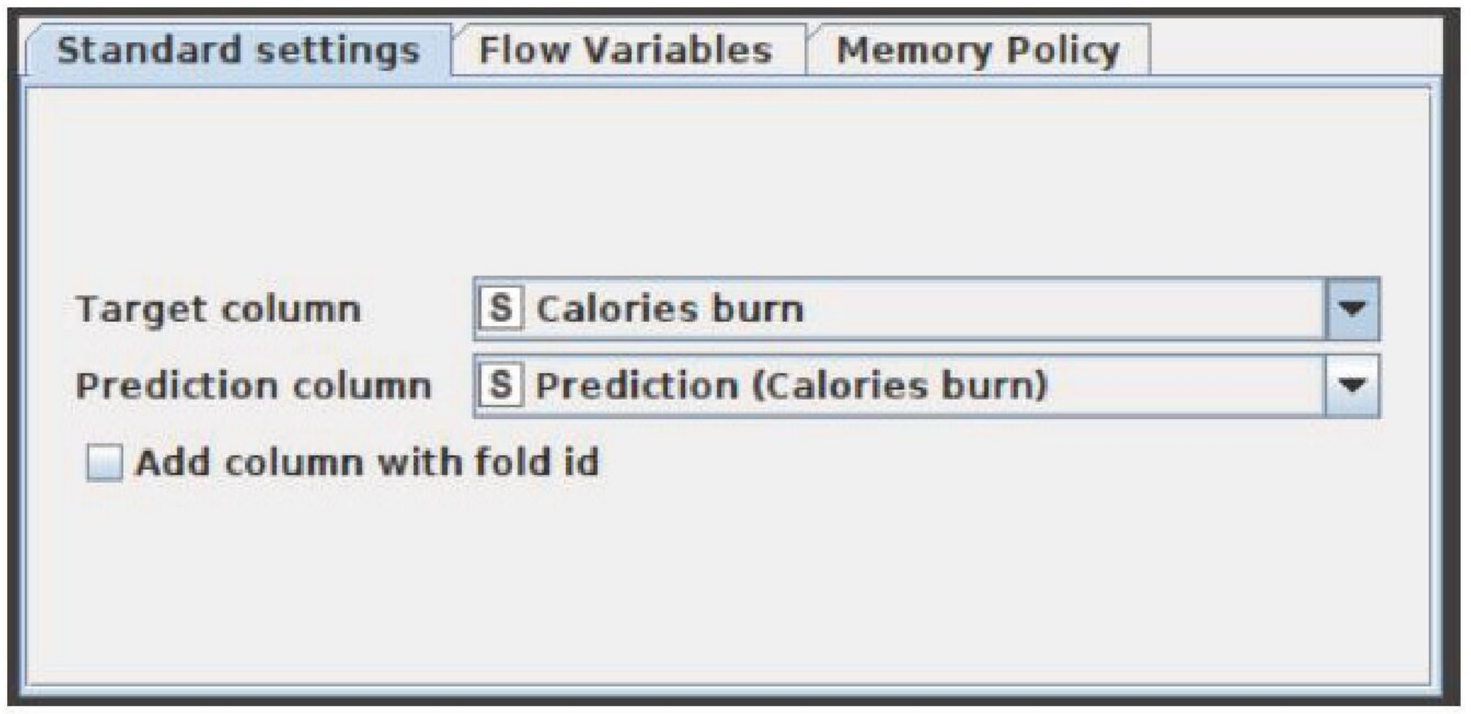


Figure 9: Configuration of X-Aggregator node

Random Forest Learner

Minitab, LLC has registered RANDOM FORESTS as a trademark, and its usage is authorised. It consists of a predetermined number of decision trees. Each decision tree model is constructed using a unique collection of rows (records), and for each branch, a random set of columns (describing attributes) is employed. Each decision tree's row sets are generated using bootstrapping and have the same size as the original input table. The attribute set for a decision tree split is determined by randomly selecting sqrt(m) attributes from the available attributes, where 'm' is the total number of learning columns. The properties may also be supplied in bit (fingerprint), byte, or double vector formats. The output model is applied to the associated predictor node and defines a random forest. This node provides a subset of the Tree Ensemble Learner's capabilities corresponding to a random forest. The configuration window for the Random Forest node that has been chosen for this application can be seen in Figure 7.

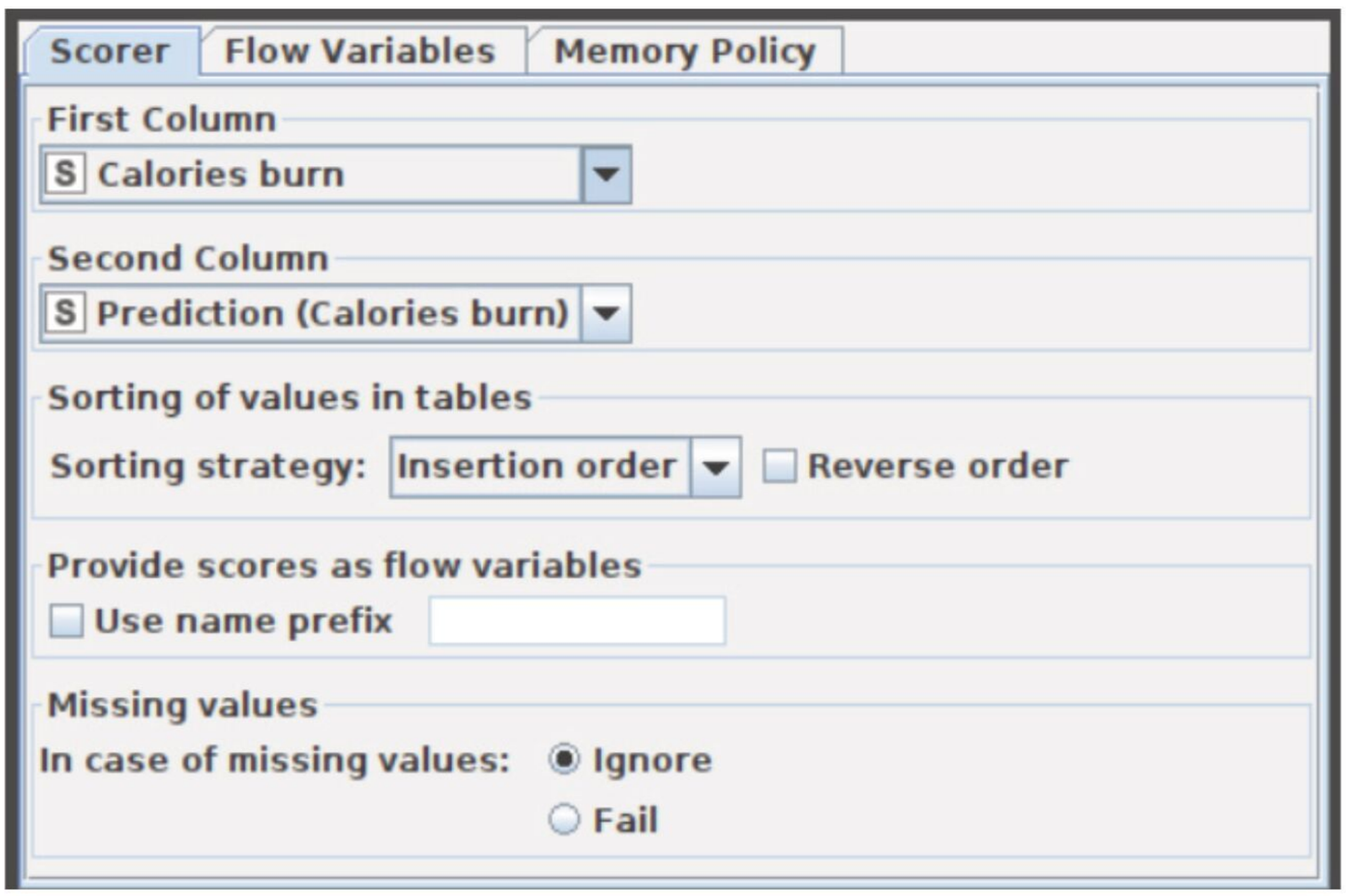


Figure 10: Configuration of Scorer node

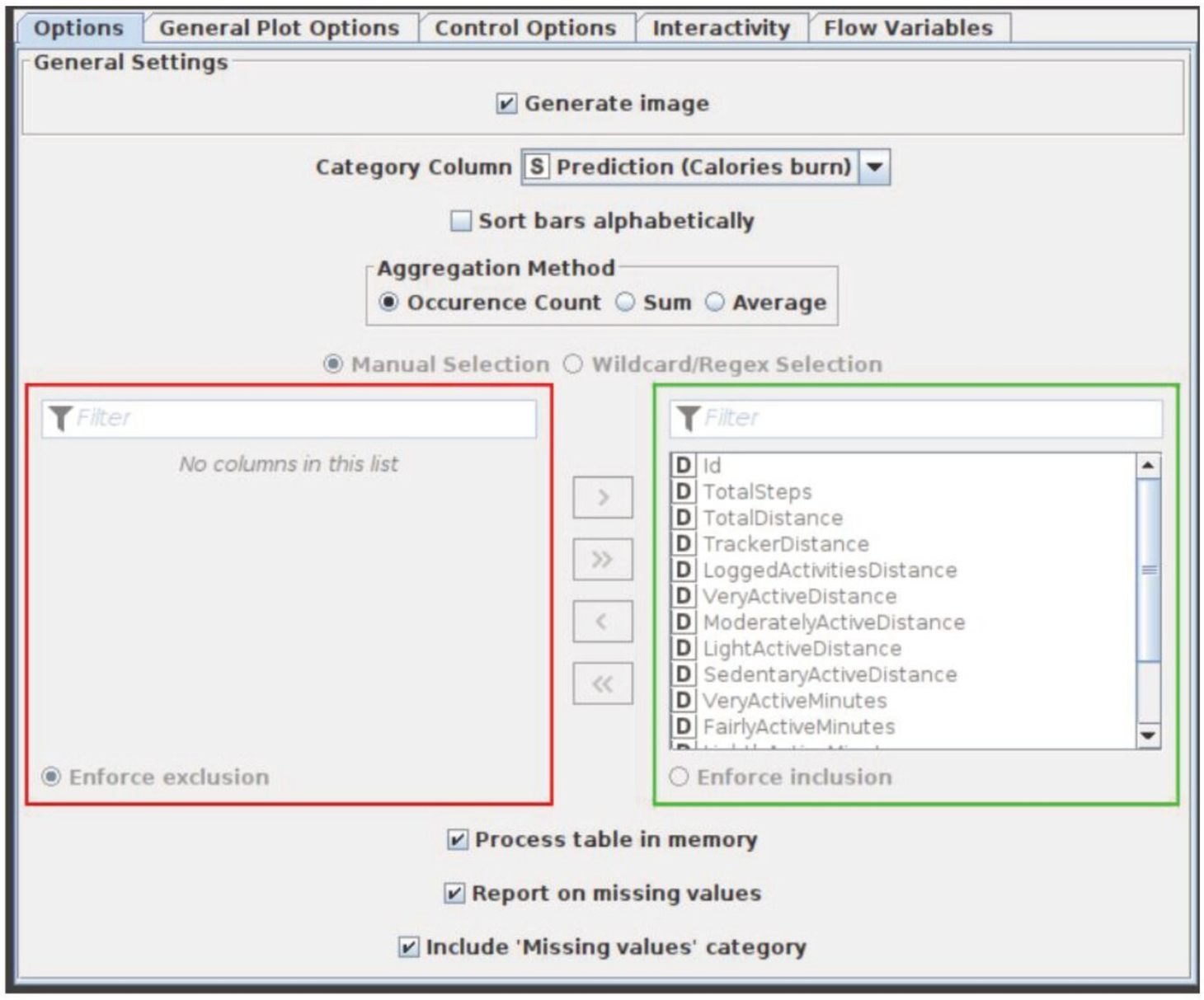


Figure 11: Configuration of Bar Plot node

Random Forest Predictor

In a random forest model, this function makes pattern predictions based on an aggregation of the predictions made by the individual trees. Figure 8 displays the settings window for the Random Forest Predictor node.

X-Aggregator

This node must be located at the conclusion of a cross-validation loop and must be the node that comes after an X-Partitioner node. It does this by first collecting the result from a predictor node, then comparing the predicted class to the actual class, and finally outputting the predictions for all rows together with the iteration statistics. Figure 9 illustrates the window for the X-Aggregator node's configuration options.

Scorer

This displays the confusion matrix (the number of rows that share an attribute with their classification) by comparing two columns based on attribute value pairs. You may also highlight cells in this matrix to reveal the underlying rows. The comparison dialogue lets you pick two columns to compare; the rows of the confusion matrix reflect the values in the first column you pick, while the columns reflect the values in the second column. The node returns a confusion matrix where each cell indicates

File Hilite

Calories ...	Low calori...	Good cal...	High calo...
Low calor...	369	0	0
Good cal...	0	234	0
High calo...	0	4	333

Correct classified: 936 Wrong classified: 4

Accuracy: 99.574% Error: 0.426%

Cohen's kappa (κ): 0.994%

Figure 12: Result of Scorer node

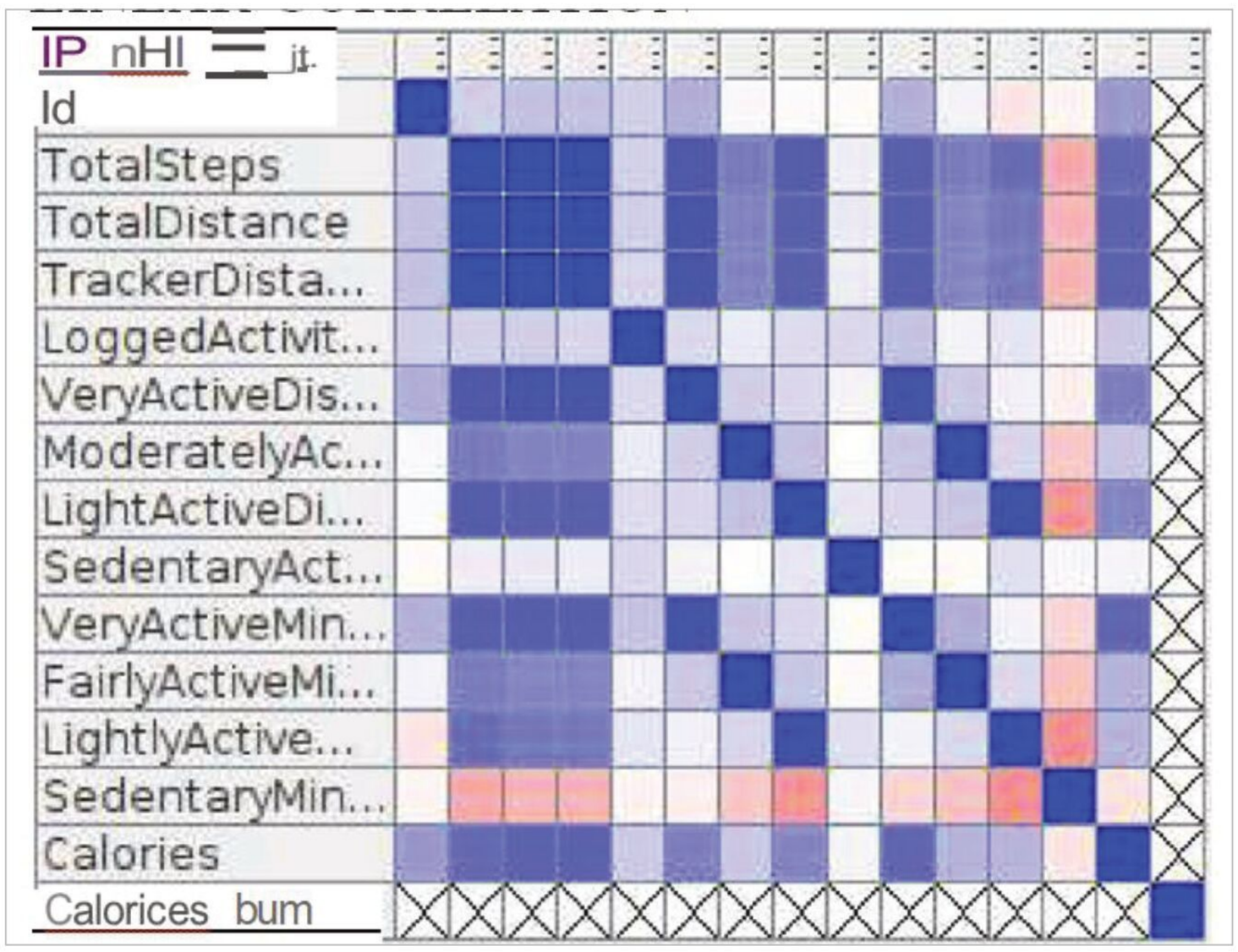


Figure 13: Result of Linear Correlation node

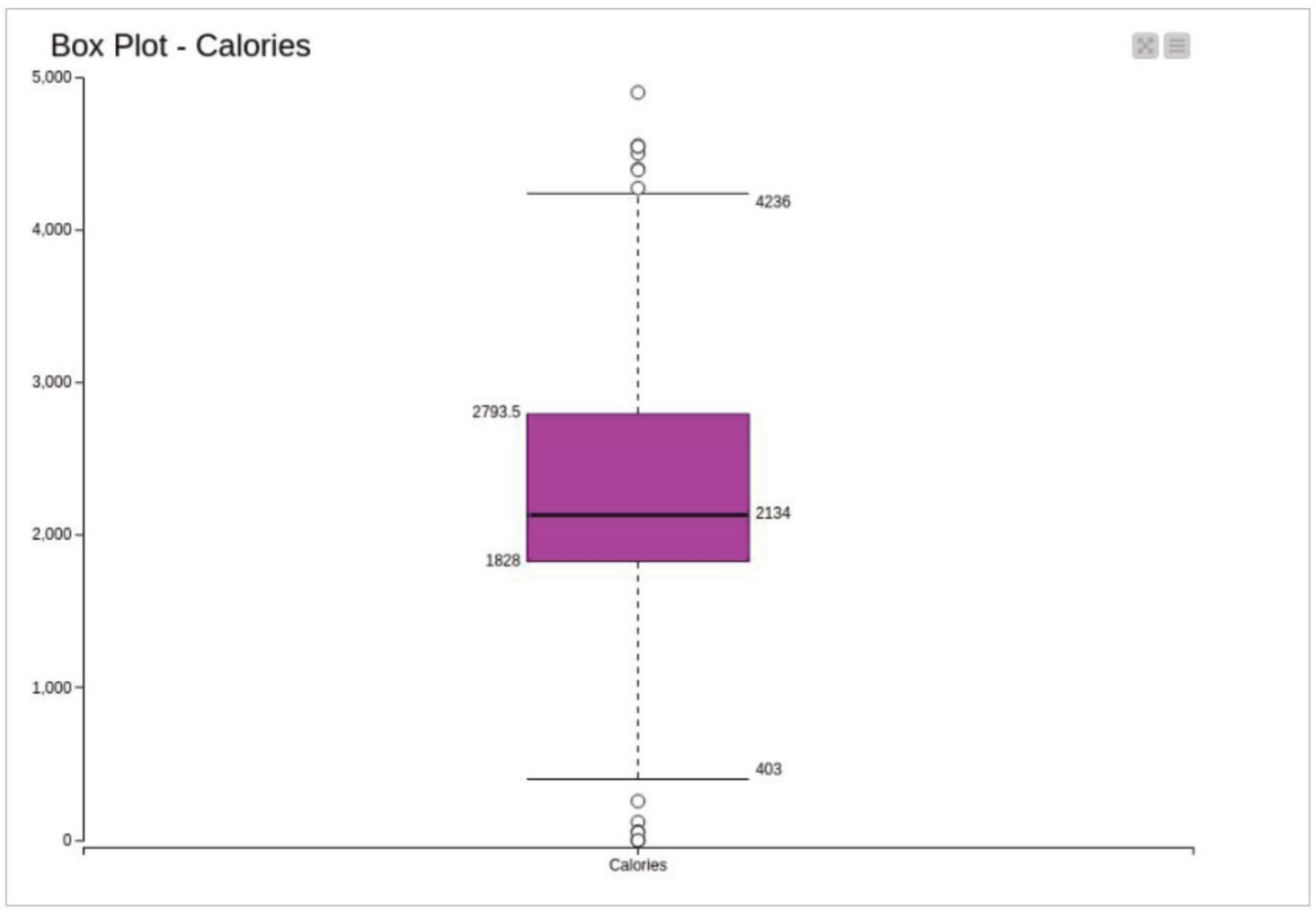


Figure 15: Box plot analysis of calories burn

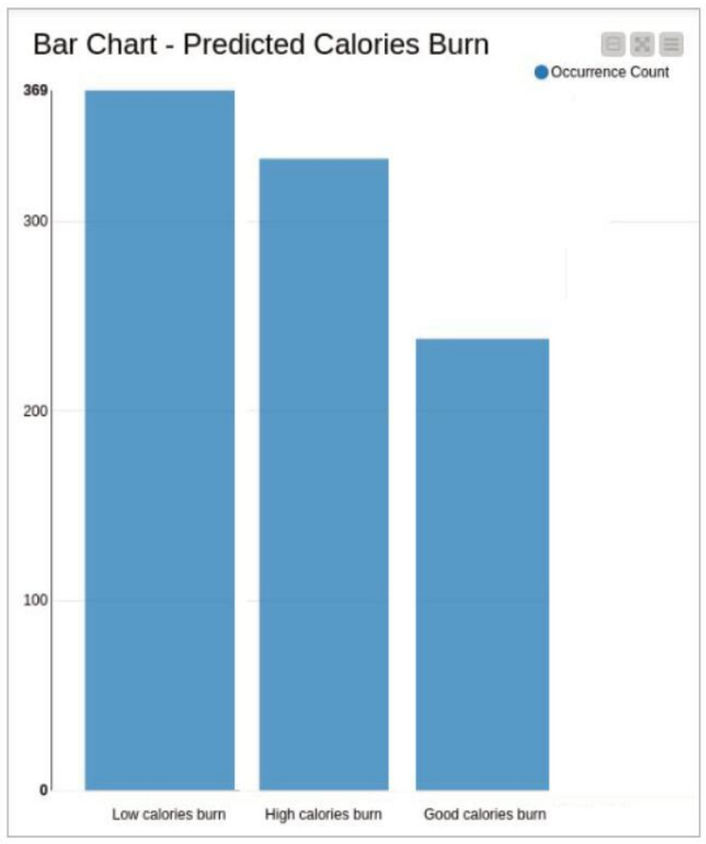


Figure 14: Bar graph analysis of calories burn

the amount of matches. Statistics on recall, precision, sensitivity, specificity, F-measure, overall accuracy, and Cohen's kappa are also reported on the second output port. The Scorer node's settings window is shown in Figure 10.

Bar Plot

Custom CSS styling is supported through a bar *char* node. The node configuration dialogue allows you to easily condense CSS rules into a single string and set it as the flow variable 'customCSS'. On KNIME's documentation page, you can discover a list of the classes that are available as well as their descriptions. Figure 11 exhibits the parameters configuration window for the Bar Plot node.

The results

Now let us see a few of the results.

Scorer node results: The prediction of the calories data set is presented in Figure 12 as the output analysis of the Scorer node.

Linear correlation output: The correlation matrix of the linear correlation analysis is presented in Figure 13.

Graphs - bar graph: Appropriate analysis has been made and the bar graph for the predicted calories burn is illustrated in Figure 14. It interprets the occurrence of low calories burn, good calories burn and high calories burn.

Row ID	D Error in %	I Size of Test Set	I Error Count
fold 0	0	94	0
fold 1	1.064	94	1
fold 2	0	94	0
fold 3	0	94	0
fold 4	0	94	0
fold 5	0	94	0
fold 6	0	94	0
fold 7	1.064	94	1
fold 8	1.064	94	1
fold 9	1.064	94	1



Figure 16: Error rate in cross-validation

Box plot: The box plot for the calories analysis is presented in Figure 15.

Error rate in cross validation : The error rate in cross validations using X-Partitioner and X-Aggregator or 10 validations is depicted in Figure 16.

The final accuracy shown by the random forest model is 99.574% and the error is 0.426%. The KNIME Analytics Platform has been successfully used to perform data analysis operations and random forest algorithm on the fitness tracker data set. END

By: Dr R. Maheswari

The author is head of the department, CSE (CPS), School of Computing Science and Engineering, Centre for Smart Grid Technologies, Vellore Institute of Technology, Chennai. She has published seven patents in her research domain. Her teaching and research expertise covers a wide range of subject areas including data analytics, machine learning, blockchain technology, embedded systems, processor level architecture, high-performance computing, reconfigurable computing, IoT, etc.

To Continued from page...52

options are floated automatically, which can be altered. A sample customisation screenshot is shown in Figure 7.

Export

The final step in the visualisation pipeline is to export the file in the needed format. The options provided by RAWGraphs are shown in Figure 8.

One of the interesting options here is to save the visualisation in '.rawgraphs' format. If you do so, this project can be opened and edited later. This format will preserve all the choices made in the earlier steps. You can share your visualisation project with others in the team so that they need not start from scratch again.

With the aforementioned data visualisation pipeline, RAWGraphs can be used to build various types of visualisation. Two more samples are provided in Figure 9 and Figure 10. Figure 9 illustrates a 'Parallel Coordinates' chart created with the 'Iris Flowers' data set. A dendrogram is shown in Figure 10. The sample data for the illustrated dendrogram can be fetched from *https://www.rawgraphs.io/learning/how-to-make-a-linear-dendrogram*. The 'Galleries' section showcases various types of illustrations built with RAWGraphs *(https://www.rawgraphs.io/gallery)*.

The official documentation of RAWGraphs is organised very well and serves as an important resource *(https://www.rawgraphs.io/learning)* to experiment with data visualisation. In addition to the excellent documentation, there are specific courses to acquire expertise *(https://www.rawgraphs.io/courses)* in using this tool.

To summarise, RAWGraphs makes the visualisation process seamless. If you are a data science enthusiast, you should give it a try. END

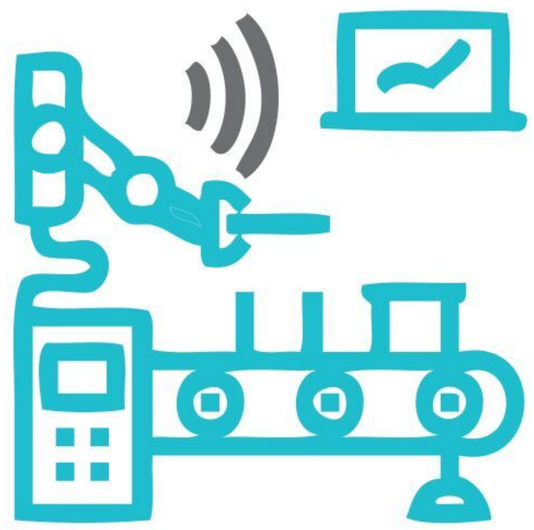
References

- RAWGraphs: *https://www.rawgraphs.io/*
- RAWGraphs documentation: *https://www.rawgraphs.io/documentation*
- RAWGraphs web app: *https://app.rawgraphs.io/*

By: Dr K.S. Kuppusamy

The author is associate professor of computer science at the School of Engineering and Technology, Pondicherry Central University, Pondicherry. He has 16+ years of teaching and research experience in academia and industry. His research interests include accessible computing, web information retrieval, mobile computing, etc.

Onboard IIoT Devices Seamlessly with the FDO Standard



Manual onboarding of a large number of Industrial Internet of Things devices into a network is cumbersome and comes with security risks. The FIDO Device Onboard (FDO) standard automates the process, making it simpler and safer.

IoT or the Internet of Things has taken the world by storm. IoT devices may include devices ranging from smart speakers, smart TVs, and even doorbells, to Industrial IoT (IIoT) devices like construction vehicles, supply chain robots, and so on.

The International Data Corporation (IDC) expects the IoT market to surpass the US$ 1 trillion mark in 2022. Industrial IoT is a major component of all IoT devices, and comprises the following.

- *Industrial things:* This contains devices like robots, PLCs (programmable logic controllers), sensors, actuators, and so on. They are generally connected to microcontrollers and microprocessors for functioning.
- *Connectivity:* As the name suggests, this is essential for providing internet or network connectivity to industrial things. It may contain switches, access points, routers, VPNs, gateways, and so on. These are responsible for creating and maintaining a network of industrial devices and connecting them to servers. Not all industrial things support Wi-Fi (IEEE 802.11) or internet connectivity. For such devices, other communication protocols like Bluetooth (RL78/G1D), Zigbee (IEEE 802.15), and LoRa WAN (IEEE 1451.0) are used.
- *Servers:* This is where the main computing, storage, and operations happen, leading to the proper functioning of the cluster or group of industrial things. Depending on the industry, the server may be hosted on a cloud platform like AWS, GCP, or Azure or on on-premises machines. It is also possible that the servers are on an offsite data centre.
- *Insights and actions:* The administrator dashboards monitor the functioning, state, and condition of the industrial things. This allows them to take action on one or multiple devices from a remote computer or mobile device. This also provides event-based alarms and alerts to the administrator in case of any issues or malfunctions.

So Industrial IoT or IIoT needs an extensive setup and onboarding procedures — connecting through the network, enrolling on the servers, adding them to the analytics dashboard, and so on. The question then is: How long would it take to manually onboard a large number, for example, 10,000 gateways, devices, and sensors? According to a 2017 study by Kaiser Associates Research and Analysis, it would take over two-man years. Yes, it is pretty cumbersome work to onboard everything manually.

Hence, the FIDO Alliance has come up with an automated and secure IoT device onboarding standard, called FIDO Device Onboard, or FDO. It is a specification for automated and secure IoT provisioning.

The FIDO Alliance, which has been working on securing services and accounts in a passwordless way, came up with the FDO specification in 2020. After some minor modifications, the FDO Specification 1.1 was proposed in 2021.

The FDO standard is currently used in various IIoT devices, mainly devices based on Intel Arm. It has proved to be one of the essential standards in the fast, secure, seamless onboarding of IoT devices. It uses cryptographic technologies to verify the ownership of an IoT device and then enrols it in the specified service seamlessly.

The FDO specification involves the use of an FDO software client installed on the IoT device when it is manufactured. A Root of Trust (RoT) key is also generated to uniquely identify each device and is saved within it, preferably within the trusted platform module (TPM) or any other secure element from which it cannot be tampered with. It can also be placed in the file system, though that is not very secure. A corresponding supply chain token is also generated, and it moves as and when the device is

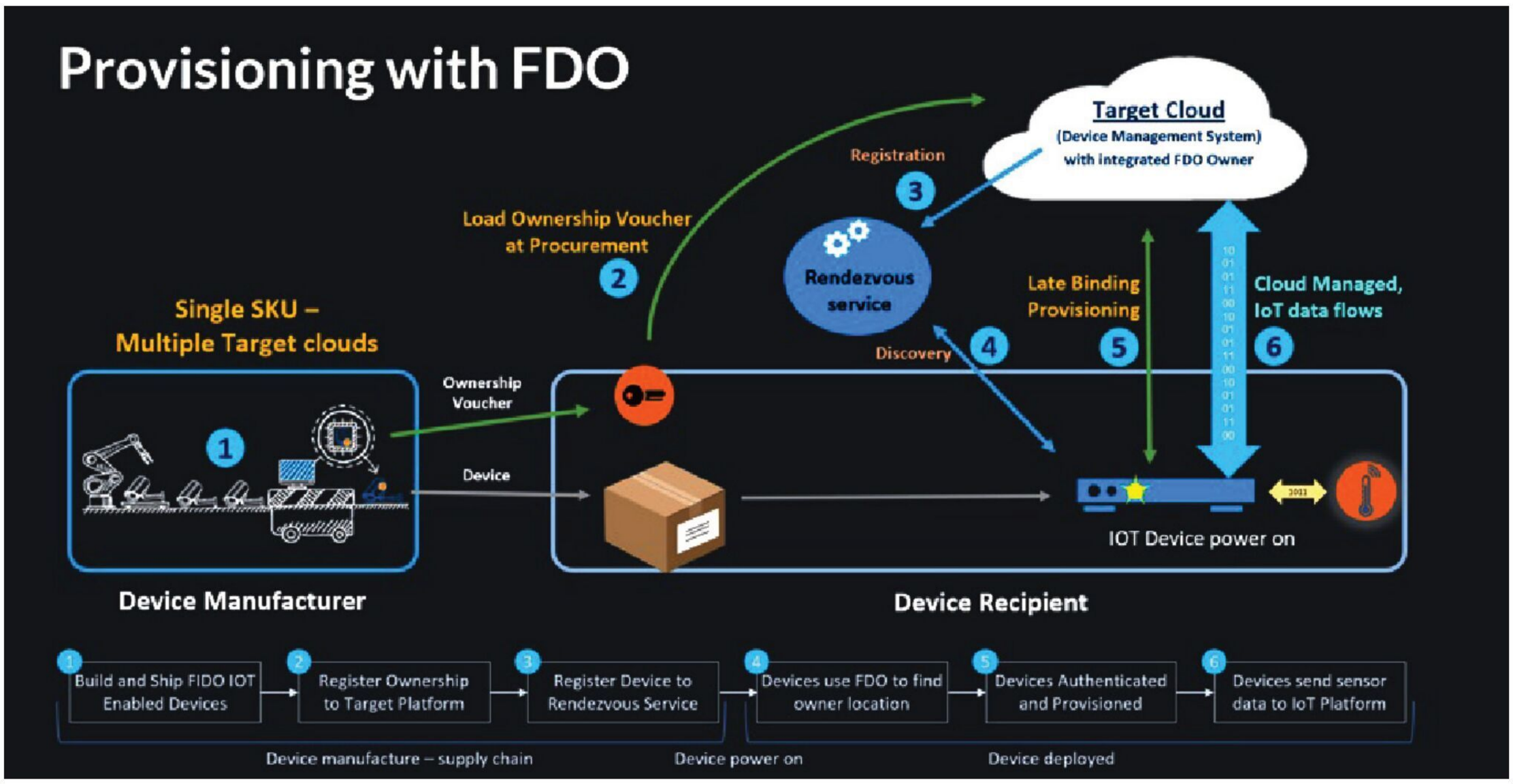


Figure 1: Provisioning IoT devices with FDO (Source: FIDO Alliance)

transferred from the manufacturer to the wholesaler, retailer, and so on, and finally to the customer. The customer enrols it on the target cloud. When the device is powered on and connected to the internet, it automatically gets provisioned taking all configurations from the target cloud.

This can be explained with the help of an example. Let us assume there is a security camera manufacturer M. This company produces five security cameras, out of which company A purchases three and company B purchases two. All five security cameras have a unique RoT key stored in their secure element. The three cameras that A purchases are sold to retailer R1 while the two B purchases go to retailer R2. When company A powers on the camera purchased, it is automatically connected to company A's server, and can be streamed by its administrator without any configuration on the camera. The same goes with the cameras procured by company B.

You may wonder how this is possible so seamlessly without any configuration on the camera. This is what happens behind the scenes. When manufacturer M manufactured the five cameras, each had a unique RoT stored in the secure element. When the devices are sold to the resellers, they use the reseller tool to generate an 'ownership token' when selling the product to the end user. Here, company A and company B are the end users. Hence, company A enrols its 'ownership tokens' on the server while company B does the same. Now when the camera is powered on, it contacts a rendezvous server, which may be of the camera manufacturer, and authenticates itself with device attestation. The rendezvous server redirects the devices to the servers of company A and company B, from where it can automatically install the configurations. This is a completely automated and seamless process, involving no manual configuration. Now, after the automated configuration is done, the device is ready to stream video feeds to the company servers.

It's pretty clear now that the FDO standard, when implemented for IoT devices, especially in industrial scenarios, can reduce human effort manifold and also provide a secure way of onboarding devices.

In the case of manual device onboarding, technicians very often get to know the credentials of the company network in order to connect devices to it. This can prove to be a major security risk if the credentials are leaked. FDO, on the other hand, uses the public key cryptosystem to securely onboard the IoT devices without any hassle or security lapses. It is ideal for corporate and industrial devices. END

By: Anisha Ghosh and Aditya Mitra

Anisha Ghosh is an open source enthusiast and a contributor to open source communities and repositories. She is interested in various development based projects.

Aditya Mitra is a cybersecurity researcher. He likes to learn about the vulnerabilities of various kinds of networks. His areas of interest are IoT, networking, and cybersecurity.

AI: Introduction to Keras and Our First Data Set

In the last article in this series on AI and machine learning, we learned to use Anaconda. We also tried to strengthen our knowledge of probability, a task we will continue in this article. We will discuss seaborn and Keras, and further explore Pandas and TensorFlow in this sixth article in the AI series.



Let us begin by increasing our theoretical knowledge of AI and machine learning. We all know that AI, machine learning, data science, deep learning, etc, are the trending topics in computer science today. However, there are other topics trending in this domain too such as blockchain, IoT (Internet of Things), quantum computing, etc. So do the advances in the field of AI affect any of these technologies in a positive way?

First, let us discuss blockchain. According to Wikipedia, "A blockchain is a type of distributed ledger technology that consists of a growing list of records, called blocks, that are securely linked together using cryptography." At a casual glance it looks as if AI and blockchain are two independent technologies advancing at a very high pace at the same time in history. But surprisingly, this is not the case. The buzzword associated with blockchain is integrity and the buzzword associated with AI is data. We give massive amounts of data to AI based applications to process. Though the results produced by many of these applications are stunning, how do we trust them? This raises the need for explainable AI, which can give certain guarantees so that end users can trust the results provided by AI based applications. Many experts believe that blockchain technologies can be used to increase the trustworthiness of decisions made by AI based software. On the other hand, smart contracts (which are part of blockchain technology) can benefit from AI based verification. Notice that, in essence, smart contracts and AI often lead to decision making. Thus, we can safely conclude that the advances in AI will positively affect blockchain technology and vice versa.

Now, let us discuss how AI and IoT impact each other. In the earlier days, IoT devices often did not have large processing power or battery backup. This made the deployment of machine learning based software, which requires large processing power, infeasible in many IoT devices. At that point in time, only rule-based AI software was being deployed in most IoT devices. The advantage with rule-based AI is that it is simple and requires relatively less processing power. However, nowadays, IoT devices are equipped with more processing power and can run powerful machine learning software. The advanced driver-assistance system developed by Tesla, called Tesla Autopilot, is an excellent example for the blending of IoT and AI. Thus, here again, AI and IoT are positively affecting each other's development.

Finally, let us try to understand how AI and quantum computing are impacting each other. Even though quantum computing is still in its infancy, quantum machine learning (QML) is taken very seriously. Quantum machine learning is based on two concepts: quantum data and hybrid quantum-classical models. Quantum data is the data generated by

a quantum computer. Quantum neural networks (QNNs) are used to model quantum computational models. A powerful tool for developing quantum computational models is TensorFlow Quantum, a library for hybrid quantum-classical machine learning. The presence of this and many other tools indicates that more and more quantum computing based AI solutions will be available in the near future.

Introduction to seaborn

Now let us learn to use seaborn. Seaborn is a Python data visualisation library based on Matplotlib. It provides functions for drawing visually appealing statistical images. Seaborn can be installed easily by using Anaconda Navigator, as explained in the previous article in this series. I have used the batting records in T20 international cricket from the website *ESPNcricinfo* to create a CSV (comma-separated values) file called *T20.csv*, which contains the following 15 columns: Player name, Career span, Matches, Innings, Not out, Total runs, Highest score, Average, Balls faced, Strike rate, Centuries, Fifties, Ducks, Number of 4s scored, and Number of 6s scored. We will use the library called Pandas to read this CSV file. Remember that we have already familiarised ourselves with Pandas. Now, consider the program shown in Figure 1.

Let us try to understand the working of the program. Lines 1 and 2 import the packages Pandas and seaborn. Line 3 reads the CSV file called *T20.csv* from the working directory of JupyterLab. Line 4 prints the metadata and the first line of data. Figure 1 shows this line, which displays the batting record of Virat Kohli, the highest run scorer in T20 international cricket matches. Line 5 stores the metadata and the first five rows of data in the CSV file *T20.csv* to another CSV file called *Best5.csv*. On execution of the code, you will see this file created in the working directory of JupyterLab. Line 6 sorts the CSV file in ascending

```
1 import pandas as pd
2 import seaborn as sns
3 df = pd.read_csv('./T20.csv')
4 print(df.head(1))
5 df.head(5).to_csv('Best5.csv')
6 df.sort_values("100", ascending=False).head(10).to_csv('Highest100.csv')
7 sns.scatterplot(data=df.iloc[:, [5, 7]].head(50), x='Ave', y='Runs')
```

```
            Player       Span  Mat  Inns  NO  Runs    HS    Ave    BF      SR  \
0  V Kohli (INDIA)  2010-2022  114   106  31  3958  122*  52.77  2865  138.15

   100  50  0   4s   6s
0    1  36  4  352  116
```

Figure 1: A simple example for seaborn

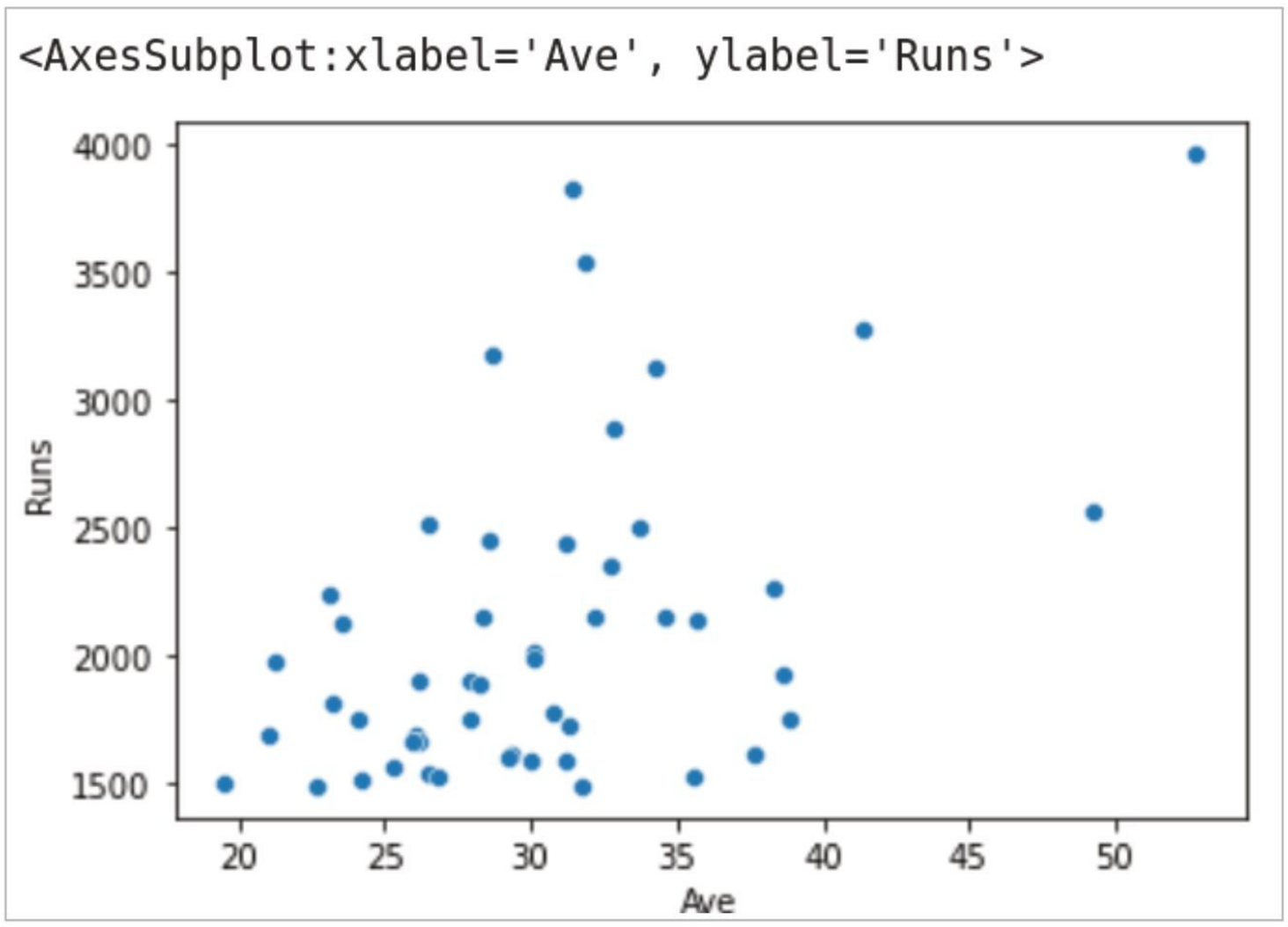


Figure 2: A scatter plot with seaborn

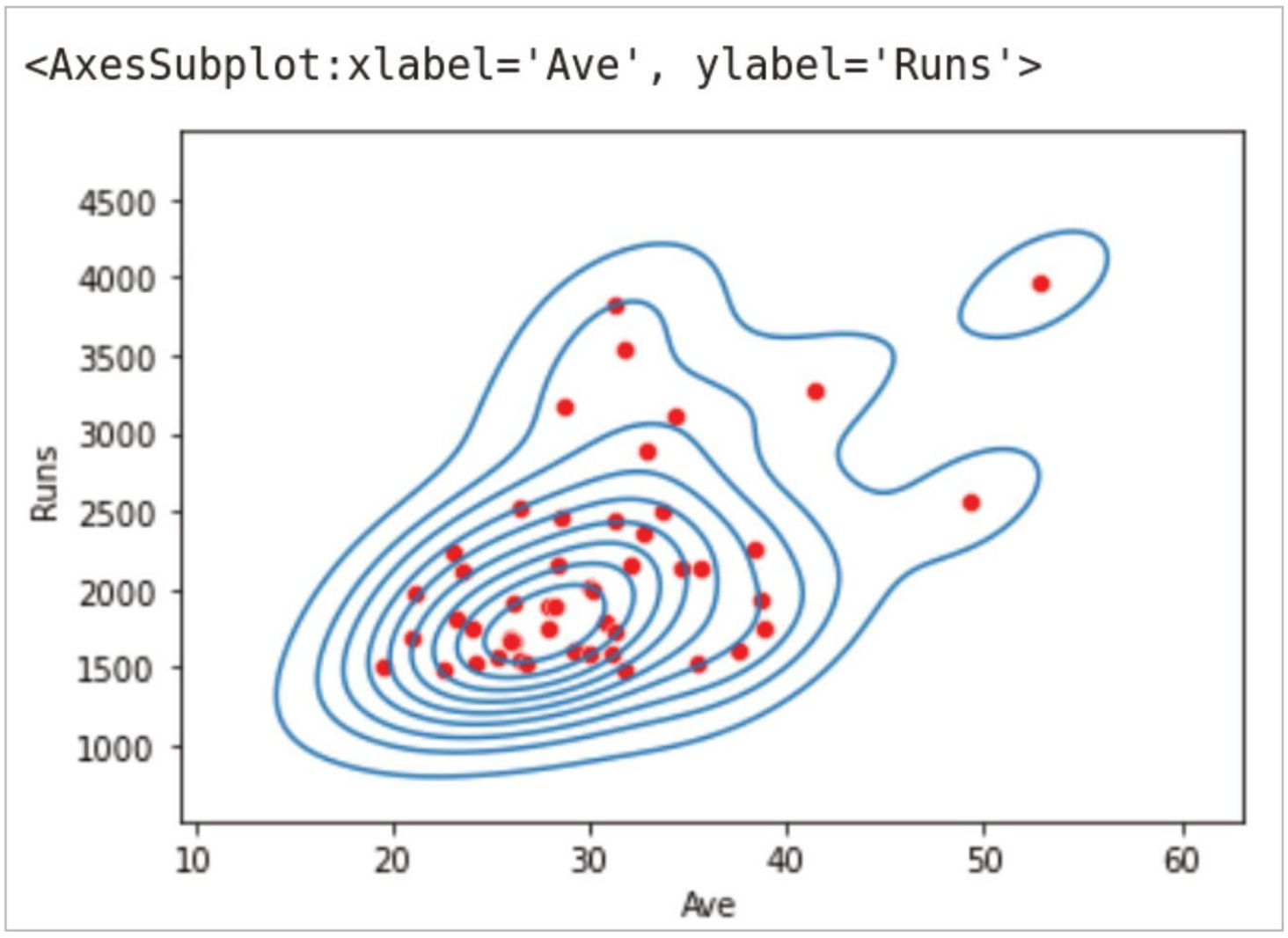


Figure 3: A KDE plot with seaborn

order based on the column '100' (which contains the number of centuries scored by each batsman) and stores the details of the top 10 century scorers into a CSV file called *Highest100.csv*. This file will also be stored in the working directory of JupyterLab. Finally, Line 7 extracts details of columns 5 and 7 (total runs scored and average, respectively) and generates a scatter plot. Figure 2 shows this scatter plot generated by the program on execution.

Now add the following line of code at the end of the program and run it again.

```
sns.kdeplot(data=df.iloc[:, [5, 7]].
head(50), x='Ave', y='Runs')
```

This draws a KDE plot, in addition to the scatter plot, of the data obtained from columns 5 and 7. The function *kdeplot( )* gives a Kernel Distribution Estimation Plot, which depicts the probability density function of the continuous or non-parametric data variables. This definition may not give you any idea about the actual operation that will be performed by the function *kdeplot( )*. Figure 3 shows the KDE plot and scatter plot being drawn on a single image. From this figure, we can observe that data points drawn by the scatter plot are grouped into clusters by the KDE plot. Notice that there are a lot of other plotting functions also provided by seaborn. In the program we have discussed now, replace Line 7 with the lines of code (one line at a time) shown below, and execute the program again. You will see plots with varying styles. Explore the other plotting functions offered by seaborn and choose the one that suits your need the most.

```
sns.histplot(data=df.iloc[:, [5, 7]].
head(50), x='Ave', y='Runs')
```

```
sns.rugplot(data=df.iloc[:, [5, 7]].
head(50), x='Ave', y='Runs')
```

```
1  from numpy import random
2  import matplotlib.pyplot as plt
3  import seaborn as sns
4
5  binom=random.binomial(n=1000, p=0.01, size=1000)
6  sns.histplot(data=binom, kde=True, color='r')
7  po=random.poisson(lam=10, size=1000)
8  sns.histplot(data=po, kde=True, color='g')
9  exp=random.exponential(size=1000)
10 sns.histplot(data=exp, kde=True, color='b')
11 plt.show()
```

Figure 4: Program for probability distributions

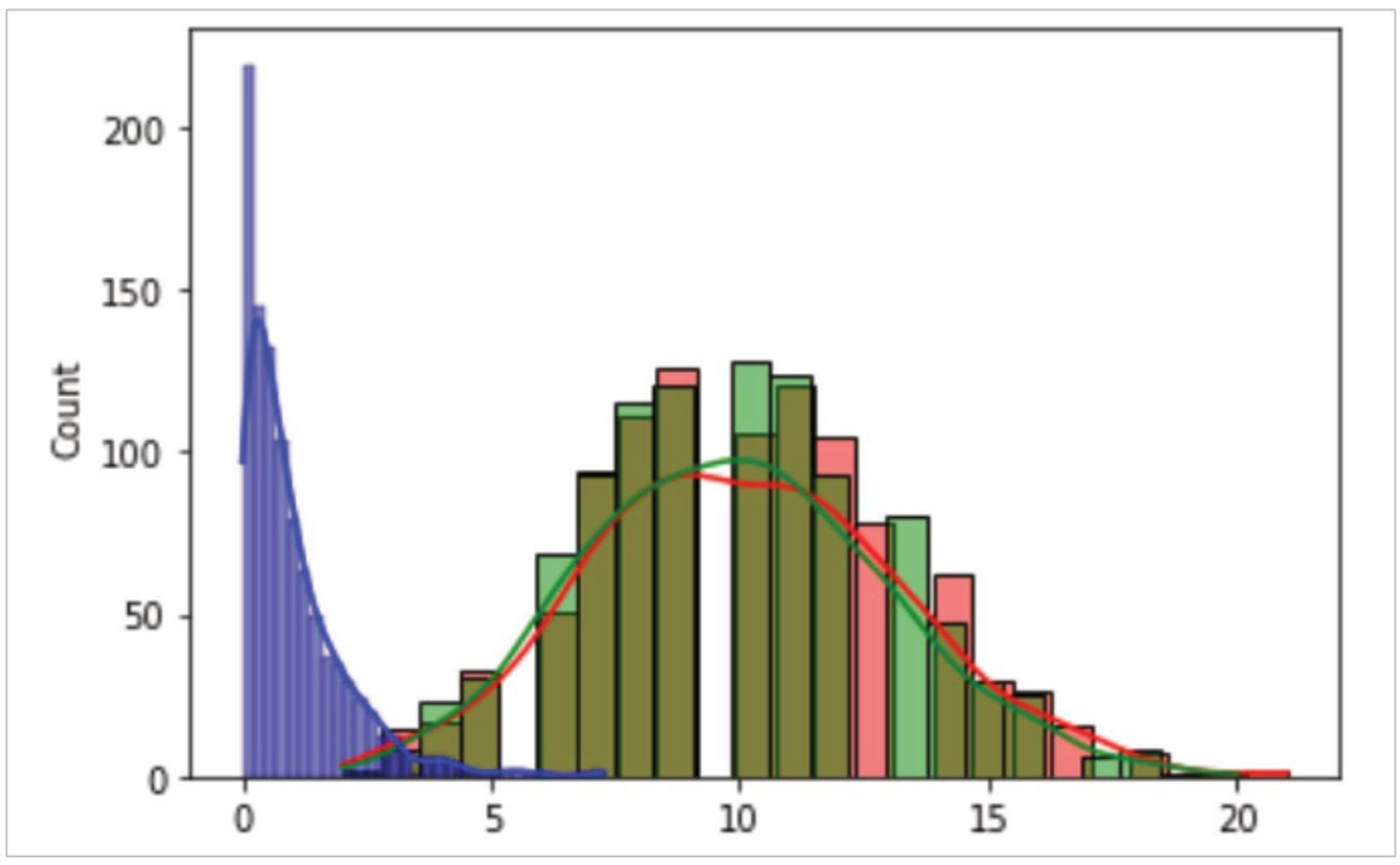


Figure 5: Plots of probability distributions

More on probability

Now, let us learn some more topics in probability. In one of the previous articles in this series, we saw how a normal distribution can be used to model real world scenarios. But normal distribution is just one among the many important probability distributions. Consider the program shown in Figure 4, which plots three probability distributions.

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  a = 10
5  b = 1
6  x = np.linspace(0,1,100)
7  y = a*x+b+np.random.randn(len(x))
8  plt.scatter(x, y, color = "r", marker = "x")
9  p = np.polyfit(x, y,1)
10 y_l = np.polyval(p,x)
11 plt.plot(x,y_l,color = "b",)
12 plt.show()
```

Figure 6: Linear regression with NumPy

Now, let us try to understand the program. Line 1 imports the *random* module of NumPy. Lines 2 and 3 import libraries Matplotlib and seaborn for plotting. Line 5 generates a binomial distribution with parameters *n* (number of trials) and *p* (probability of success). Binomial distribution is a discrete probability distribution, which gives the number of successes in a sequence of *n* independent experiments. The third parameter called *size* decides the shape of the output. Line 6 plots the histogram of the data generated. Notice that a KDE plot is also plotted because the parameter *kde* is set as true. The third parameter *colour* decides the colour of the plot, which in this case is red ('*r*'). Line 7 generates a Poisson distribution (named after the French mathematician Siméon Denis Poisson). The Poisson distribution is a discrete probability distribution which gives the limit of the binomial distribution. The parameter *lam* sets the expected number of events occurring in a fixed-time interval. Here also the parameter *size* decides the shape of the output. Line 8 plots the data generated as a histogram in green. Line 9 generates an exponential distribution with a size of 1000. Line 10 plots the data generated as a histogram in blue. Finally, Line 11 displays all the plots generated. Figure 5 shows the three probability distributions plotted on execution of the program. The *random* module of NumPy offers a large number of other probability distributions like Dirichlet distribution, Gamma distribution, geometric distribution, Laplace distribution, etc. It would be really rewarding if you could work and get familiarised with more of them.

Now, let us learn about linear regression. Linear regression analysis can be used to predict the value of a variable based on the value of another variable. One of the important applications of linear regression is

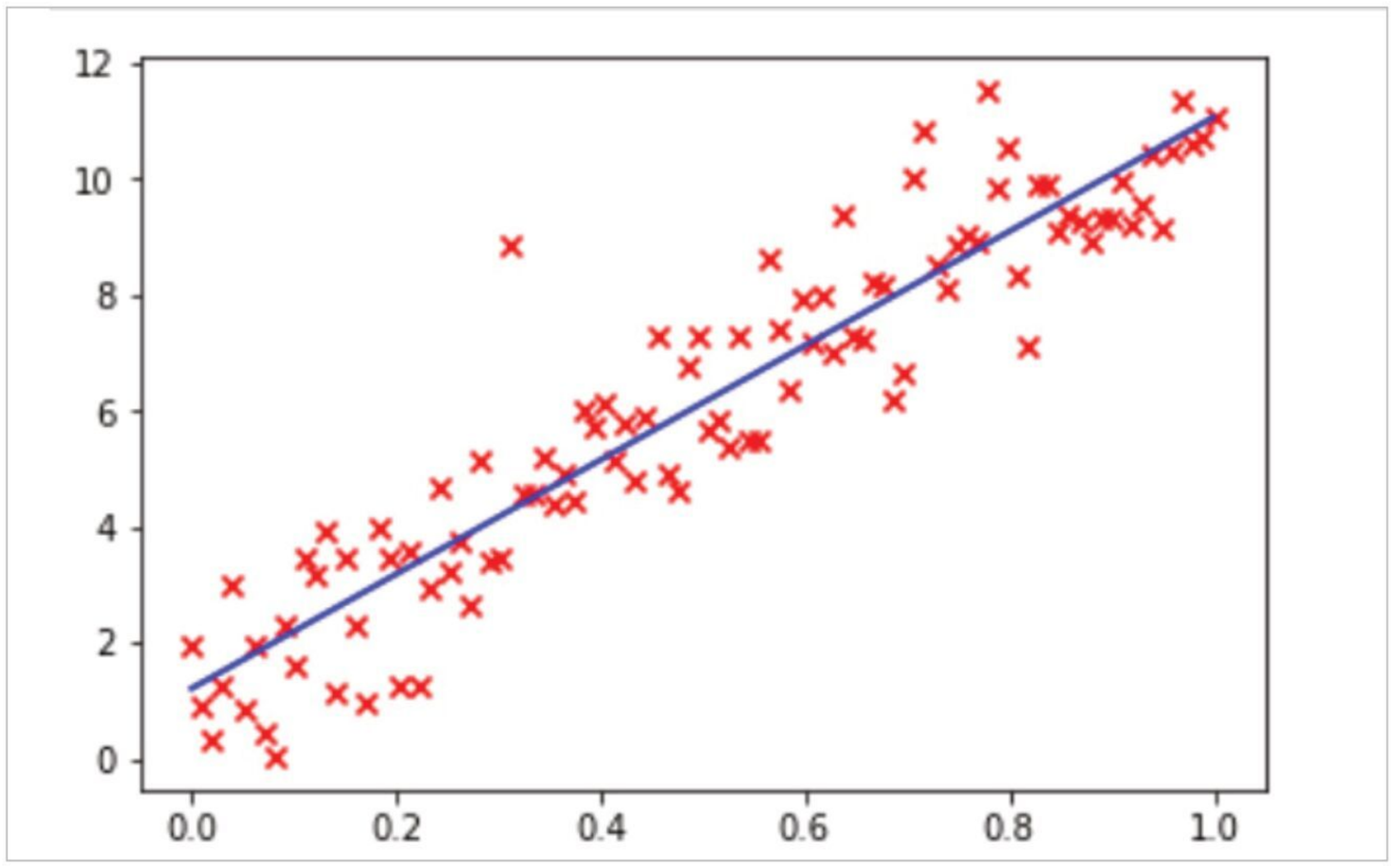


Figure 7: Linear regression scatter plot 1

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy import stats
4  a = 10
5  b = 1
6  x = np.linspace(0,1,100)
7  y = a*x+b+np.random.randn(len(x))
8  plt.scatter(x, y, color = "y", marker = "o")
9  m, c, r, p, std_err = stats.linregress(x, y)
10 y_l = list(m*x+c)
11 plt.plot(x,y_l,color = "g",)
12 plt.show()
```

Figure 8: Linear regression with SciPy

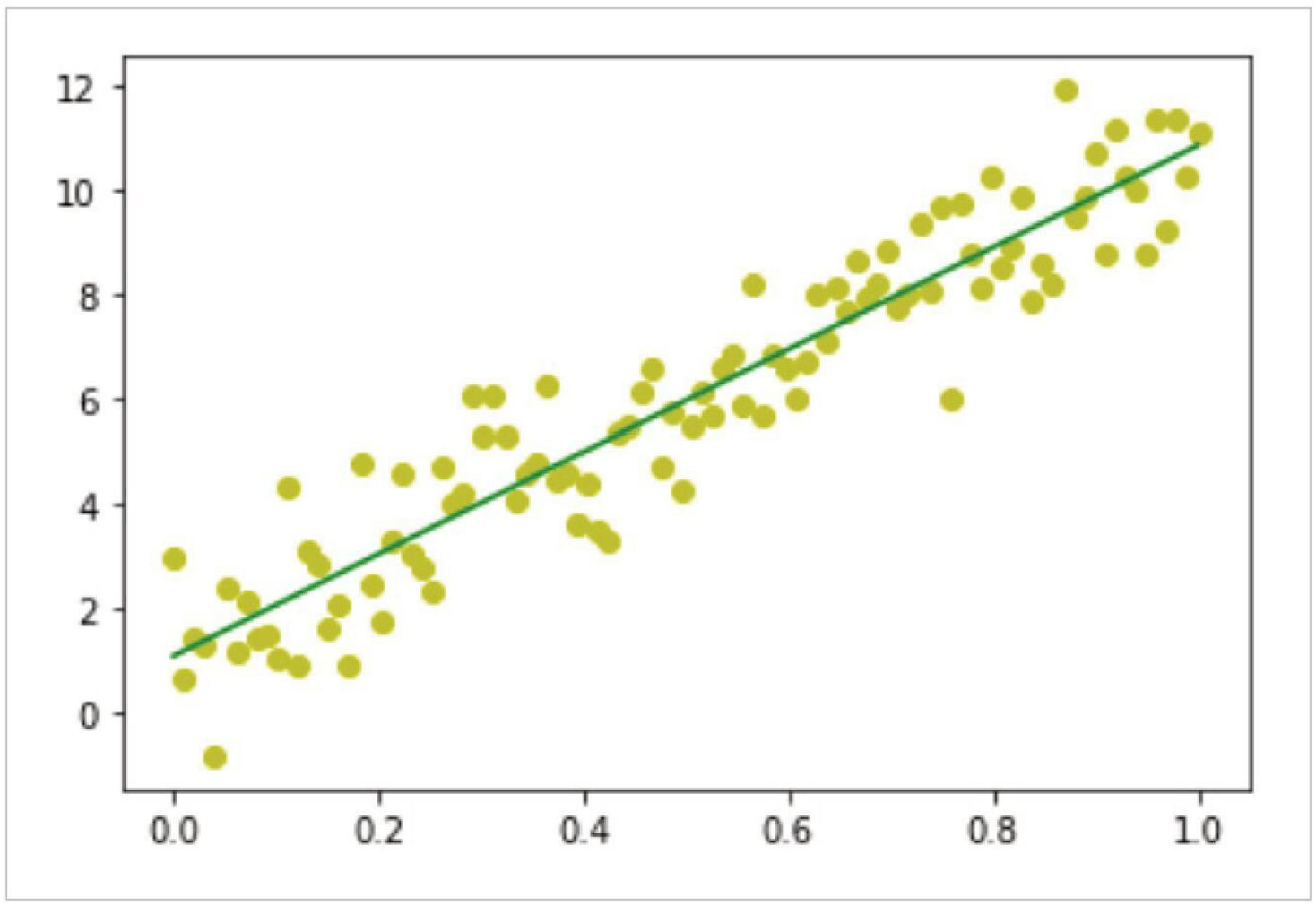


Figure 9: Linear regression scatter plot 2

```
1  import tensorflow as tf
2  x = tf.constant([-111.0, -222.0, 0.0, 333.0, 444.5], dtype=tf.float32)
3  print(tf.nn.relu(x))
4  print(tf.nn.leaky_relu(x))
```

```
tf.Tensor([  0.    0.    0.  333.  444.5], shape=(5,), dtype=float32)
tf.Tensor([-22.2 -44.4   0.  333.  444.5], shape=(5,), dtype=float32)
```

Figure 10: TensorFlow for neural networks

```
1  from tensorflow import keras
2  from keras.datasets import mnist
3  import matplotlib.pyplot as plt
4
5  (train_imgs, train_lbls), (test_imgs, test_lbls) = mnist.load_data()
6  print(train_imgs.ndim)
7  print(train_imgs.shape)
8  print(test_imgs.ndim)
9  print(test_imgs.shape)
10 sample = train_imgs[1234]
11 plt.imshow(sample, cmap=plt.cm.binary)
12 plt.show()
```

Figure 11: Our first data set

data fitting. Linear regression is very important because of its simplicity. The supervised learning paradigm of machine learning is actually another name for regression modelling. Thus, linear regression can be considered as an important machine learning strategy. This learning paradigm is often called statistical learning by statisticians. Both NumPy and SciPy offer functions for performing linear regression. Since linear regression is an important operation in machine learning, we will see examples involving both NumPy and SciPy. Now, consider the program shown in Figure 6.

The program uses NumPy to perform linear regression. First, let us try to understand the working of this program. Lines 1 and 2 import NumPy and Matplotlib. Lines 4 and 5 initialise the variables *a* and *b* for data generation. Line 6 uses the function *linspace( )* to generate evenly spaced numbers over the specified interval of 0 and 1. In this case 100 such numbers are generated. Line 7 generates values in the array *y* by using the values in the array *x*. The function *randn( )* returns samples from the standard normal distribution. These values, together with the values provided by the variables *a*, *b*, and array *x*, are used to generate the values in the array *y*. Line 8 generates a scatter plot by using the values in the arrays *x* and *y*. Figure 7 shows this scatter plot. The hundred data points are marked in red. Line 9 uses the function *polyfit( )* to perform a least squares polynomial fit. It is one of the techniques used to perform linear regression. The function *polyfit( )* takes for processing the arrays *x* and *y*, and a third parameter (in this case 1), which denotes the degree of the fitting polynomial. The function then returns the polynomial coefficients and they are stored in the array *p*. Line 10 uses the function *polyval( )*, which evaluates a polynomial at specific values. These values are stored in the array *y_l*. Line 11 plots the regression line in blue (Figure 7). Finally, Line 12 displays all the plots. This regression line can be used for making predictions about the possible *(x, y)* data pairs.

Now, let us see how SciPy can be used to perform the same task. Consider the program shown in Figure 8.

Let us try to understand the working of the program. Lines 1 and 2 import the libraries NumPy and Matplotlib. Line 3 imports the *stats* module from the library SciPy. Lines 4 to 8 are the same as the previous program and perform similar tasks. Line 9 uses the function *linregress( )* offered by the stats module of SciPy, which can calculate a linear least-squares regression for two sets of

measurements — in this case, the values in the arrays *x* and *y*. The function returns, in addition to other data, the slope and intercept of the regression line that is stored in the variables *m* and *c*, respectively. Line 10 uses the values of slope and intercept to generate the regression line. Line 11 plots the regression line in green. Finally, Line 12 displays all the plots. Figure 9 shows the image generated on execution of the program. The data points are displayed in yellow and the regression line is shown in green.

In the last three articles in this series, we have tried to learn more and more concepts from probability and statistics. Though the treatment of the subject is in no way comprehensive, I think we have arrived at a good starting point and it is time to shift our focus to other matters of equal importance.

Introduction to Keras

Now, let us learn to use Keras. Keras is used along with TensorFlow. Hence, let us begin by exploring TensorFlow a bit more. Consider the program shown in Figure 10. Though this program contains just four lines of code, this is our first encounter with neural networks in this series. Now, let us try to understand the working of the program. Line 1 imports the library TensorFlow. Line 2 creates a tensor called *x*. Lines 3 and 4 apply two activation functions called ReLU (Rectified Linear Unit) and Leaky ReLU, respectively, on the tensor *x*. In neural networks, the activation function of a node defines the output of that node given an input or set of inputs. ReLU activation function is an activation function defined as the positive part of its argument. The output of the code in Line 3 is shown in Figure 10. It can be observed that the negative values in the tensor *x* are replaced with zeros after applying the ReLU activation function. Leaky ReLU activation function is a modified version of the ReLU activation function. From the output of the code in Line 4 shown in Figure 10, it can be verified that positive values are retained as such, and 20 per cent of the negative values are retained on applying the Leaky ReLU activation function. Though we are moving on to discuss Keras, we will revisit this topic in the future to learn more about neural networks and activation functions.

Now, let us start using Keras. The installation of Keras can also be done very easily with Anaconda Navigator. Consider the program shown in Figure 11. We will use our first data set in this program. For the time being, we will only import and display sample data from the data set. But in the next article, we will use the same data set for training and testing our model, thus beginning the next phase in our journey to develop AI and machine learning based applications.

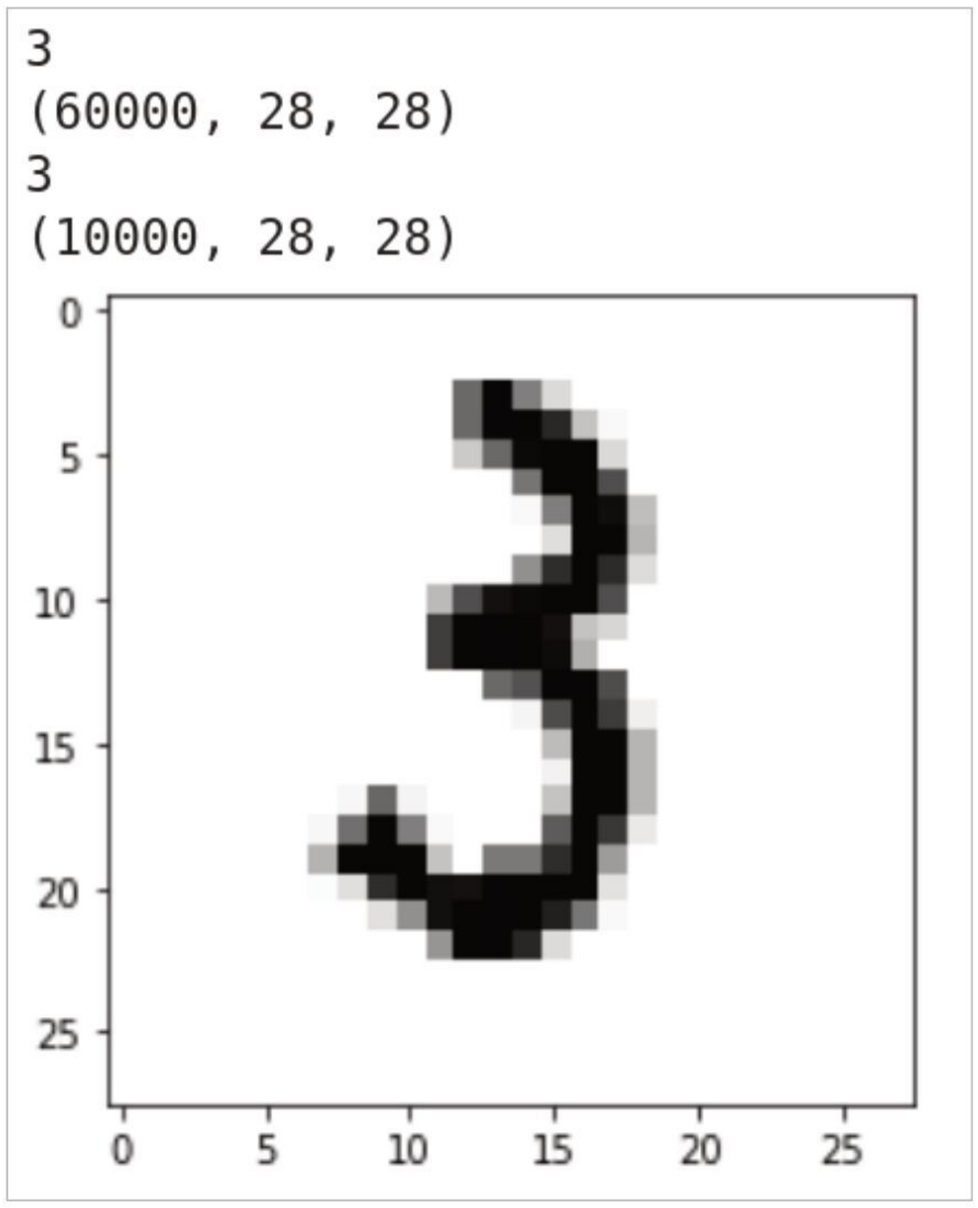


Figure 12: A sample from the data set

Now, let us try to understand the working of the program. Line 1 imports Keras. Line 2 imports the MNIST database of handwritten digits. It has a training data set of 60,000 samples and a test data set of 10,000 samples. Line 3 imports Matplotlib. Line 5 loads the data set. Lines 6 to 9 print the dimension and shape of the training data and the test data. Figure 12 shows the output of these lines of code. It can be seen that both the training data and the test data are three-dimensional. It can also be verified that there are 60,000 training images of 28 x 28 pixel quality and 10,000 test images of 28 x 28 pixel quality. Line 10 loads the 1234th training image. Lines 11 and 12 display this image. Figure 12 shows this handwritten image of the digit 3.

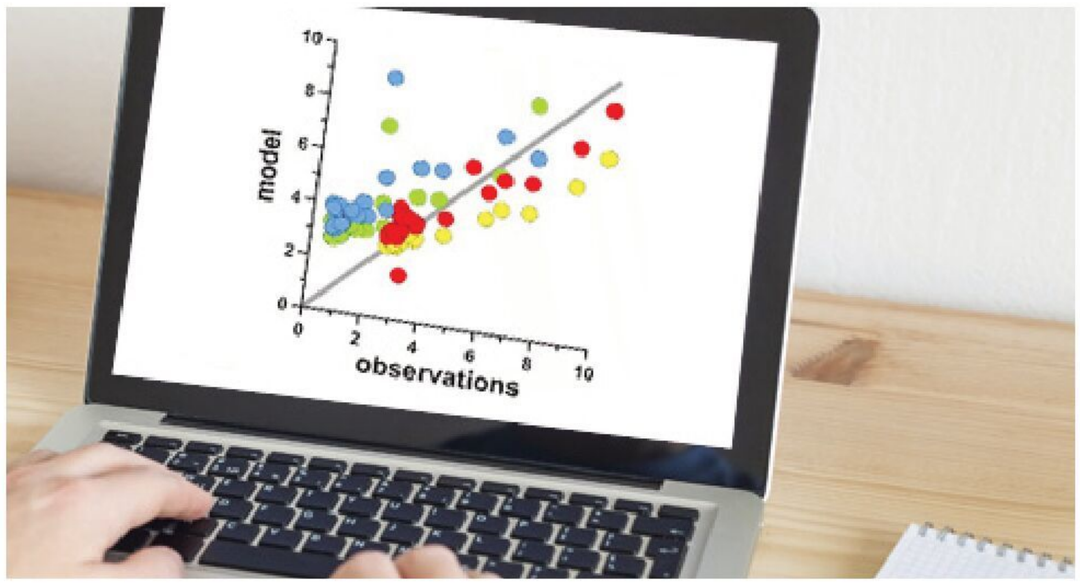
Now, it is time to wind-up our discussion. We are at the halfway mark of our journey through AI and machine learning. We have started discussing neural networks, a discussion we will continue in the next article. We have also come across our first data set with Keras. In the next article, we will continue discussing the use of Keras by training with this data set. We will also have our first interaction with scikit-learn, another powerful Python library for machine learning. END

By: Deepu Benson

The author is a free software enthusiast. His area of interest is theoretical computer science. The open source tools of his choice include ns-2 and ns-3. He maintains his own technical blog too.

R Series: Correlation

This sixteenth article in the R series will introduce you to correlation.



In this article, we shall explore correlation. We will use R version 4.2.1 installed on Parabola GNU/Linux-libre (x86-64) for the code snippets.

```
$ R --version

R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under the terms of the
GNU General Public License versions 2 or 3.
For more information about these matters see
https://www.gnu.org/licenses/.
```

The *cor()* function in R can compute the correlation between two sets of vectors. Its usage is as follows:

```
cor(x, y, na.rm, use, method)
```

The correlation function accepts the following arguments:

Argument	Description
x	numeric vector, matrix or data frame
y	vector or NULL
na.rm	logical value to remove missing values
use	string to specify the computing method
method	'pearson', 'kendall', or 'spearman' coefficient

Let us create three vectors 'x', 'y' and 'z' for comparison using the *sin()* and *cos()* functions as follows:

```
> t = seq(0, 10, 0.1)
> x = sin(t)
> y = sin(t + 0.05)
> z = cos(t)
```

The first few values of the vectors are listed below:

```
> head(x)
[1] 0.00000000 0.09983342 0.19866933 0.29552021 0.38941834
0.47942554

> head(y)
[1] 0.04997917 0.14943813 0.24740396 0.34289781 0.43496553
0.52268723

> head(z)
[1] 1.0000000 0.9950042 0.9800666 0.9553365 0.9210610
0.8775826
```

The correlation between the 'x' and 'y' vectors as well as the 'x' and 'z' vectors is shown below:

```
> cor(x, y)
[1] 0.9985339

> cor (x, z)
[1] 0.05483627
```

We observe that there is a high correlation of 0.99 between the 'x' and 'y' vectors as they come from the same sine function. The 'x' and 'z' vectors have a low correlation of 0.05 as they are defined using sine and cosine functions respectively.

Consider the *mtcars* data set available in the *lattice* library:

```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

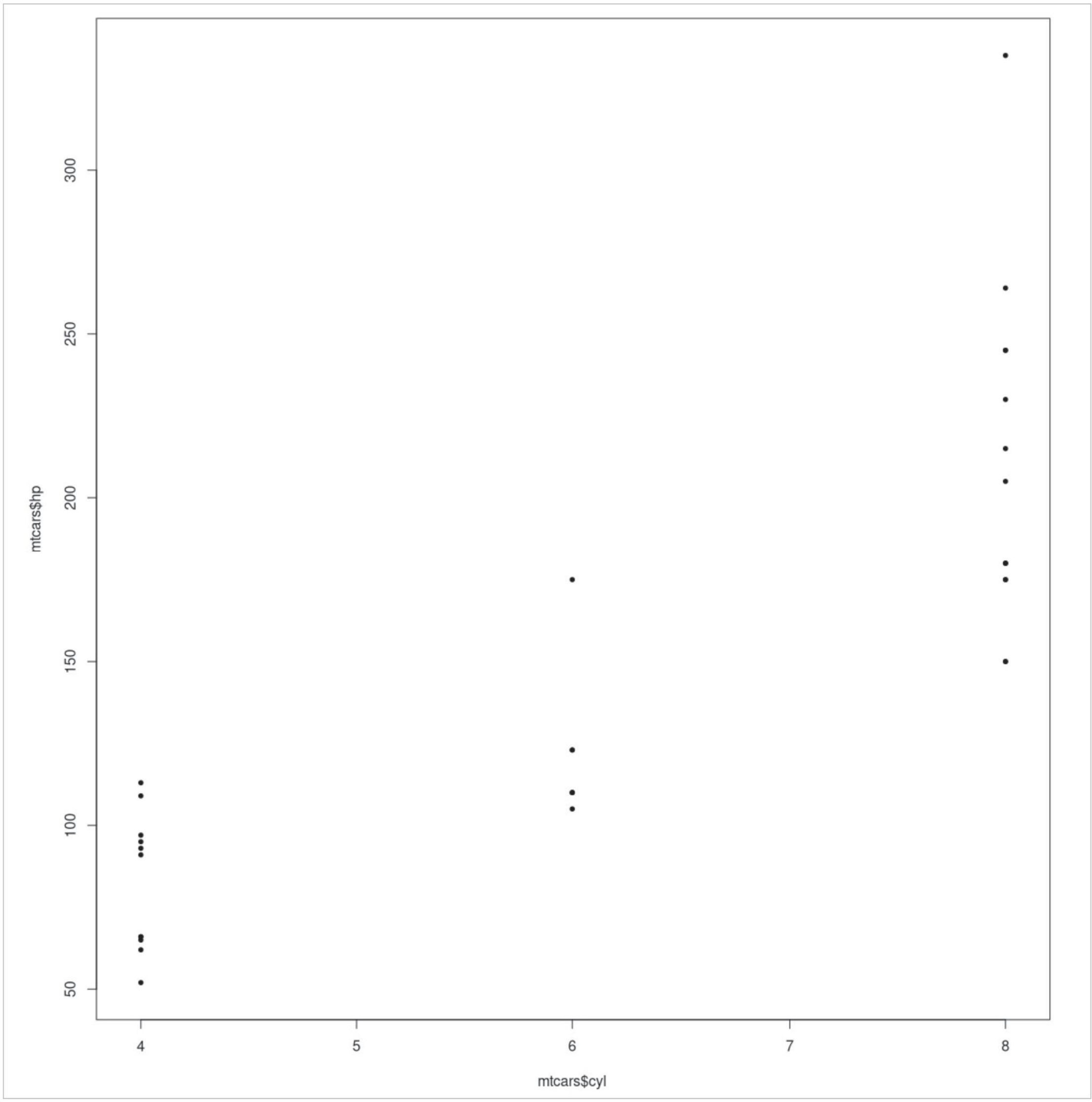


Figure 1: Plot *mtcars disp* and *hp*

A plot comparison between cylinder size and horsepower can be generated using the *plot()* function, as follows:

```
> plot(mtcars$cyl, mtcars$hp, pch=20)
```

The correlation coefficient values between the cylinder size and horsepower using the default Pearson's, Kendall's and Spearman's methods are given below:

```
> cor(mtcars$cyl, mtcars$hp)
[1] 0.8324475

> cor(mtcars$cyl, mtcars$hp, method = "kendall")
[1] 0.7851865

> cor(mtcars$cyl, mtcars$hp, method = "spearman")
[1] 0.9017909
```

The high correlation coefficient signifies that a high horsepower has a positive relation with the number of

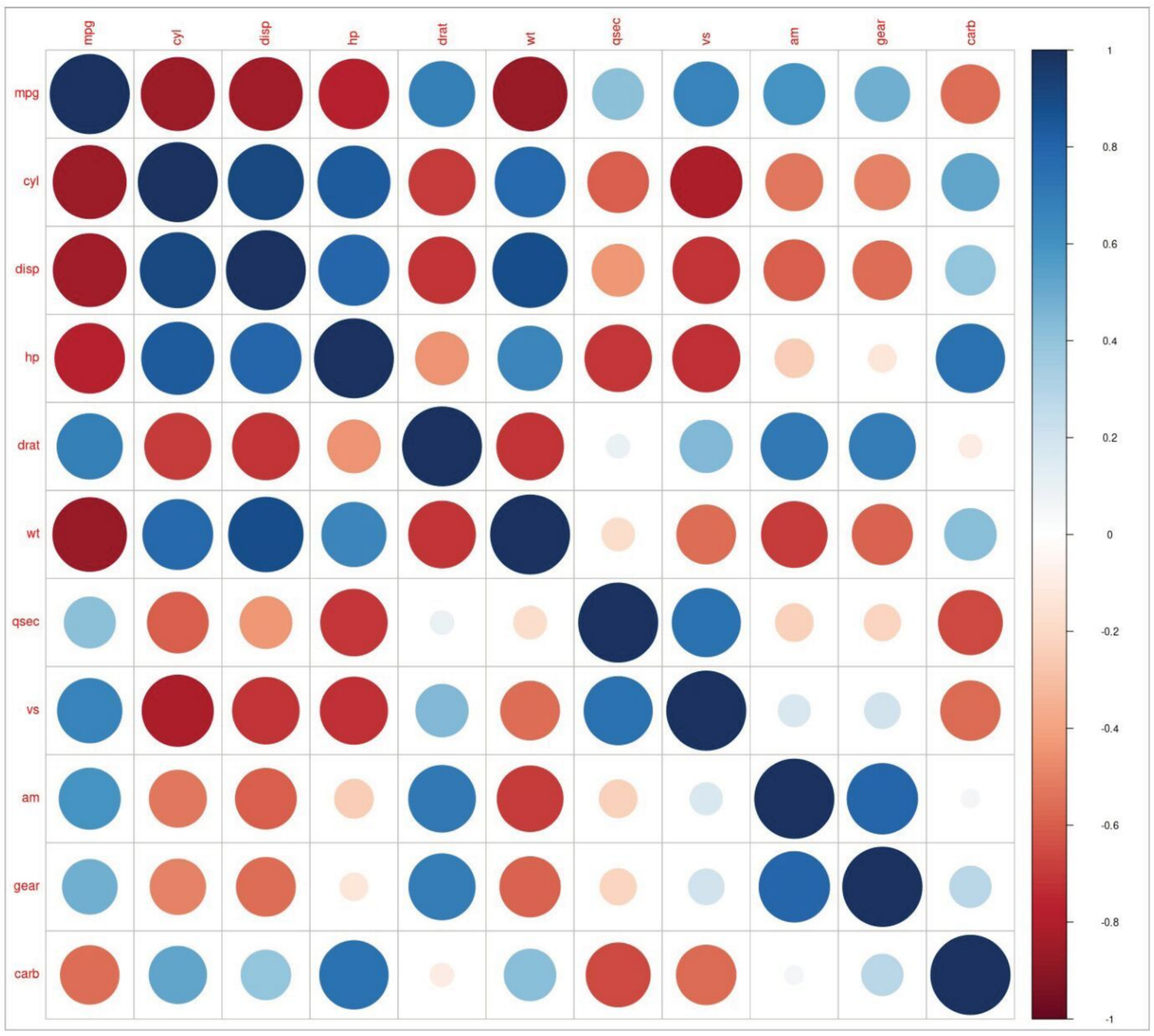


Figure 2: *Corrplot*

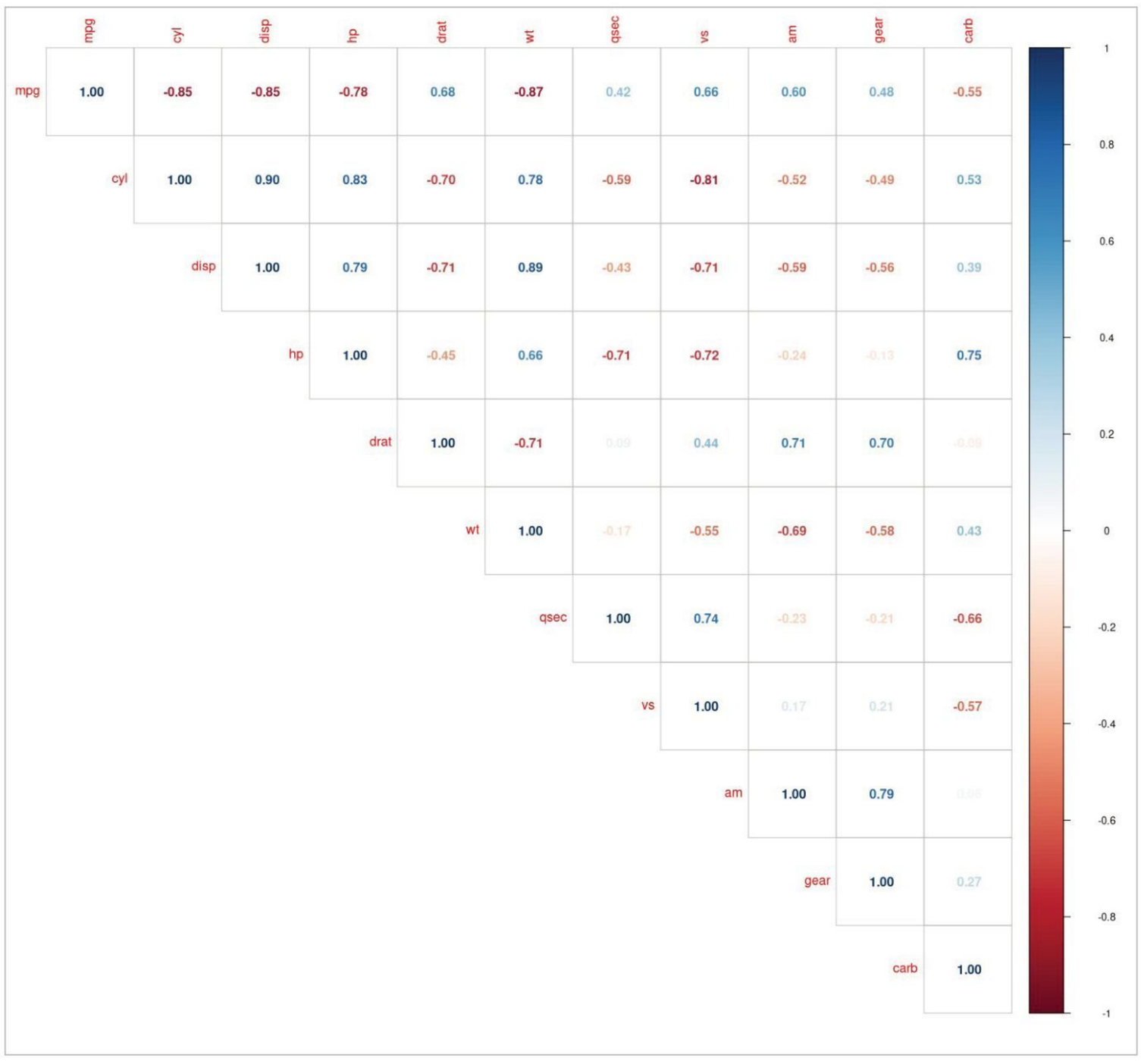


Figure 3: Corrplot (upper segment)

cylinders. The *cor.test()* function can also be used to test the association between paired samples. The correlation test between the *mtcars* cylinders and horsepower values is shown below:

```
> cor.test(mtcars$cyl, mtcars$hp)

        Pearson's product-moment
correlation

data:  mtcars$cyl and mtcars$hp
t = 8.2286, df = 30, p-value = 3.478e-09
alternative hypothesis: true correlation
is not equal to 0
95 percent confidence interval:
 0.6816016 0.9154223
sample estimates:
      cor
0.8324475
```

The *cor.test* function accepts the following arguments:

Argument	Description
x, y	numeric data vectors
alternative	'two.sided', 'greater' or 'less' alternative hypothesis
method	'pearson', 'kendall' or 'spearman'
exact	logical value to be indicated if exact p-value should be computed
conf.level	confidence level
continuity	true to use a continuity correction
data	optional data frame or matrix
subset	optional vector that specifies subset of observations to be used
na.action	function to indicate when data has NA values

We can also handle missing values in the data source vectors by specifying the 'use' argument with the *cor()* function.

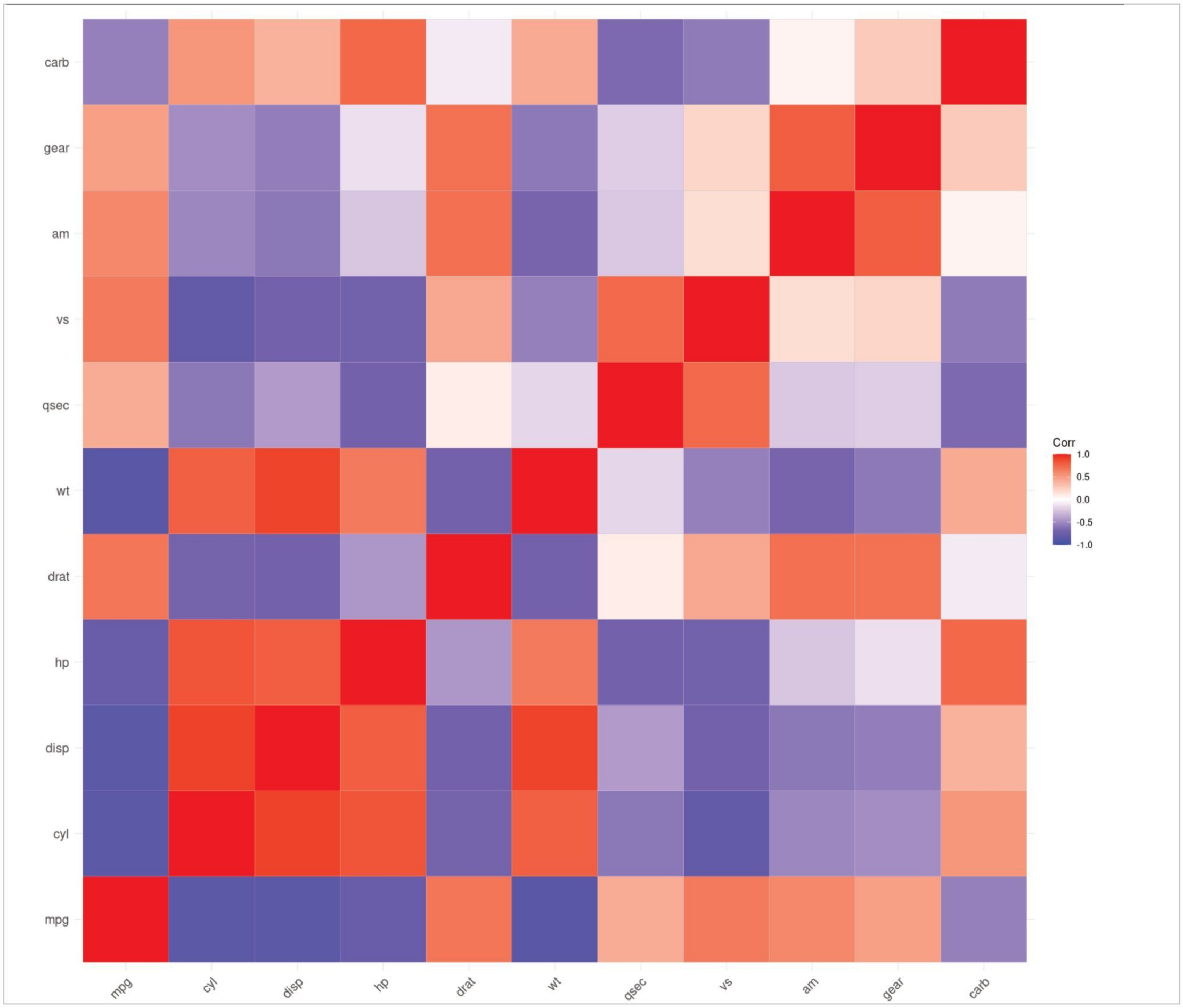


Figure 4: *ggcorrplot*

An example is given below:

```
> a <- c(1, 3, 5)
> b <- c(2, 4, NA)

> cor(a, b)
[1] NA

> cor(a, b, use = "complete.obs")
[1] 1
```

An MxN correlation matrix can be created for a data frame. For example:

```
> cor(mtcars)
            mpg        cyl       disp         hp       drat         wt
mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.68117191
-0.8676594
cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.69993811
0.7824958
disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.71021393
0.8879799
hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.44875912
0.6587479
drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.00000000
-0.7124406
wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.71244065
1.0000000
qsec  0.4186840 -0.5912421 -0.4336979 -0.7082234  0.09120476
-0.1747159
vs    0.6640389 -0.8108118 -0.7104159 -0.7230967  0.44027846
-0.5549157
am    0.5998324 -0.5226070 -0.5912270 -0.2432043  0.71271113
```

```
-0.6924953
gear  0.4802848 -0.4926866 -0.5555692 -0.1257043  0.69961013
-0.5832870
carb -0.5509251  0.5269883  0.3949769  0.7498125 -0.09078980
0.4276059

           qsec         vs          am       gear        carb
mpg   0.41868403  0.6640389  0.59983243  0.4802848 -0.55092507
cyl  -0.59124207 -0.8108118 -0.52260705 -0.4926866  0.52698829
disp -0.43369788 -0.7104159 -0.59122704 -0.5555692  0.39497686
hp   -0.70822339 -0.7230967 -0.24320426 -0.1257043  0.74981247
drat  0.09120476  0.4402785  0.71271113  0.6996101 -0.09078980
wt   -0.17471588 -0.5549157 -0.69249526 -0.5832870  0.42760594
qsec  1.00000000  0.7445354 -0.22986086 -0.2126822 -0.65624923
vs    0.74453544  1.0000000  0.16834512  0.2060233 -0.56960714
am   -0.22986086  0.1683451  1.00000000  0.7940588  0.05753435
gear -0.21268223  0.2060233  0.79405876  1.0000000  0.27407284
carb -0.65624923 -0.5696071  0.05753435  0.2740728  1.00000000
```

The *corrplot()* function can be used to display a correlation matrix. You can install the same in an R session using the following command:

```
> install.packages("corrplot")

Installing package into '/home/guest/R/x86_64-pc-linux-gnu-
library/4.1'
...
 ** testing if installed package can be loaded from temporary
location

 ** testing if installed package can be loaded from final
location

 ** testing if installed package keeps a record of temporary
installation path

 * DONE (corrplot)
```

After loading the *corrplot* library, we can view the plot for the *mtcars* data set as follows:

```
> library("corrplot")
corrplot 0.92 loaded

> corrplot(cor(mtcars), method = "circle")
```

You can also restrict the plot to the upper segment by using the 'type' argument. For example:

```
 > corrplot(cor(mtcars), method = "number", type = "upper")
```

The *corrplot()* function accepts the following arguments:

Argument	Description
corr	the correlation matrix
method	'circle', 'square', 'ellipse', 'number', 'pie', 'shade' and 'colour'
type	'full', 'upper', or 'lower'
col	specifies a vector colour of glyphs
bg	background colour
title	title of the graph
add	logical value to add plot to an existing graph
diag	logical value to display the correlation coefficients
order	'original', 'AOE', 'FPC', 'hclust', or 'alphabet'
rect.col	colour for the rectangular border
tl.cex	size of the text label
t1.col	colour of the text label
tl.srt	numeric value for text label string rotation

Another plotting function for the correlation matrix is the *ggcorplot()* function, as illustrated below:

```
> install.packages("ggcorrplot")

Installing package into '/home/shakthi/R/x86_64-pc-linux-gnu-
library/4.1'

...
 ** testing if installed package can be loaded from final
location

 ** testing if installed package keeps a record of temporary
installation path
 * DONE (ggcorrplot)


> library("ggcorrplot")
Loading required package: ggplot2
```

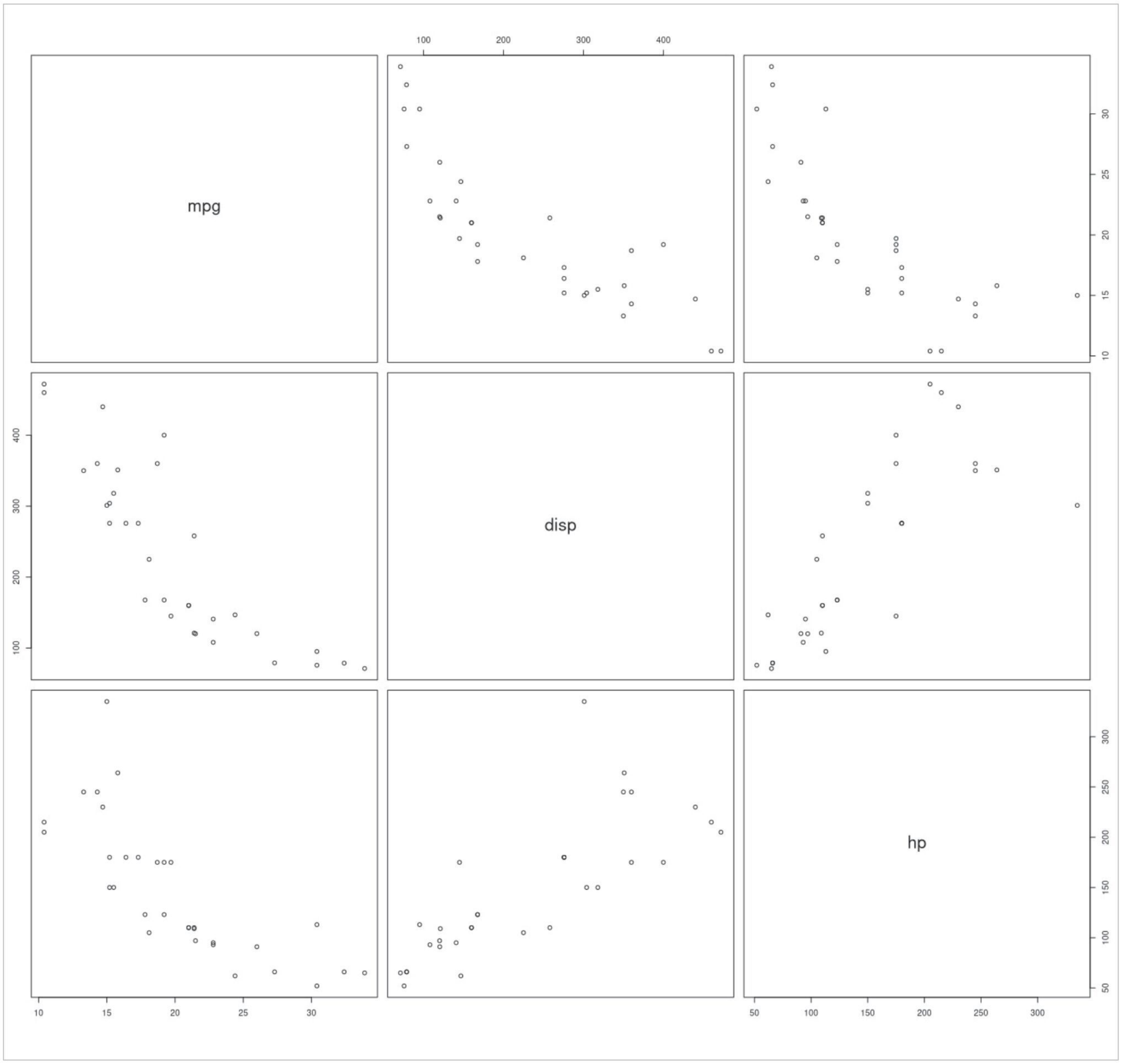


Figure 5: pairs

```
> ggcorrplot(cor(mtcars))
```

The scatterplots are also useful for visualising the matrix. The *pairs()* function is used to compare the miles per gallon, displacement and horsepower, as shown below:

```
> pairs(mtcars[, c("mpg", "disp", "hp")])
```

The *ggscatterstats()* function accepts a data frame, and produces a combined density and histogram plot. It is provided by the *ggstatsplot* library, which is an extension of the *ggplot2* package. An example is given below:

```
> install.packages("ggstatsplot")
 ...
 ** testing if installed package keeps a record of
temporary installation path

 * DONE (ggstatsplot)

> library(ggstatsplot)

> ggscatterstats(data = mtcars, x = cyl, y = hp)
```

The *ggscatterstats()* function accepts the following arguments:

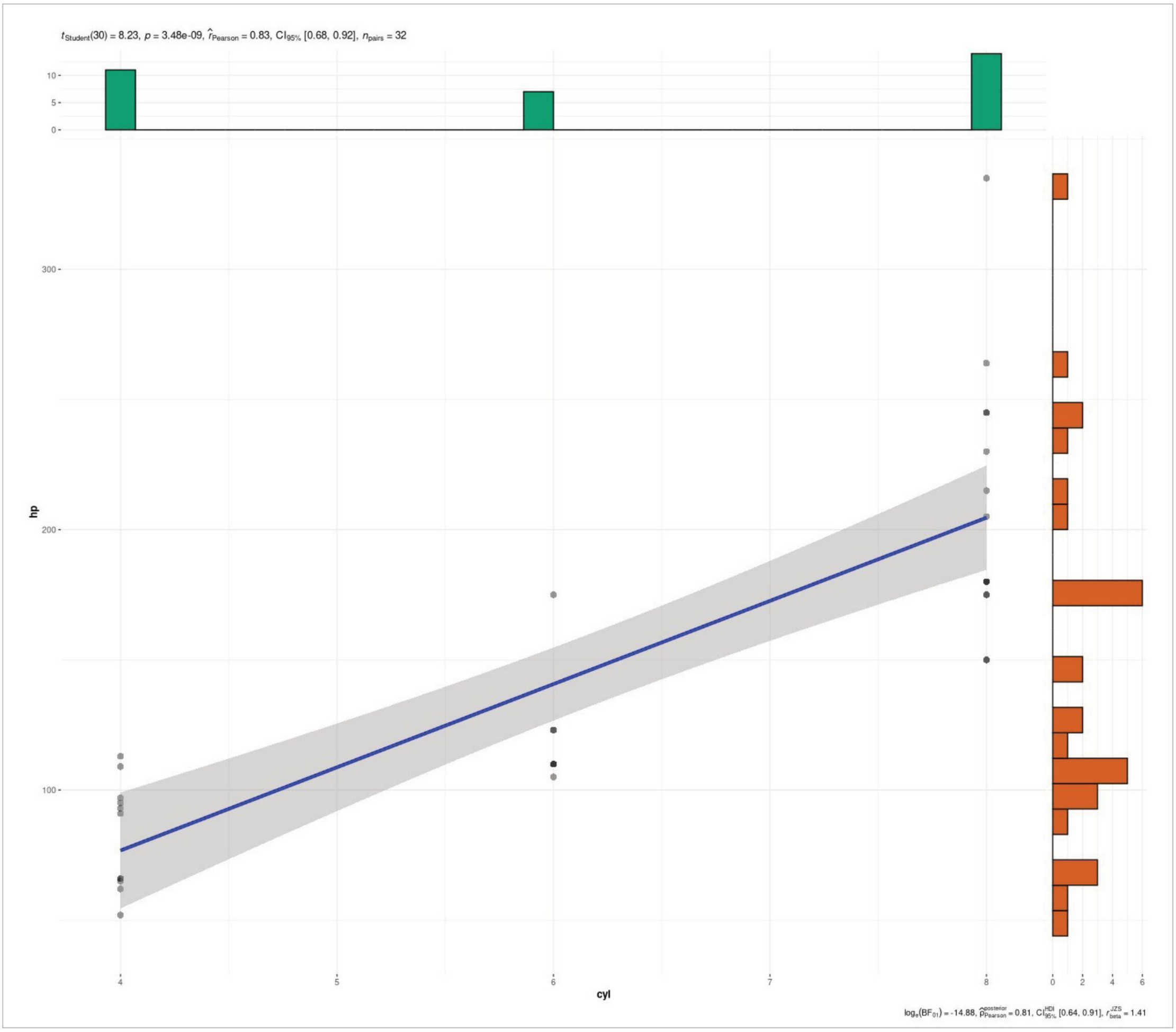


Figure 6: ggscatterstats

Argument	Description
data	data frame or matrix, table, array
x	explanatory variable in the data
y	response variable in the data
type	'parametric', 'nonparametric','robust',' bayes'
bf.prior	prior width for calculating Bayes factors
bf.message	logical value to display Bayes Factor
tr	trim level for the mean
k	significant digits after decimal point
xfill, yfill	colour fill for x and y axes

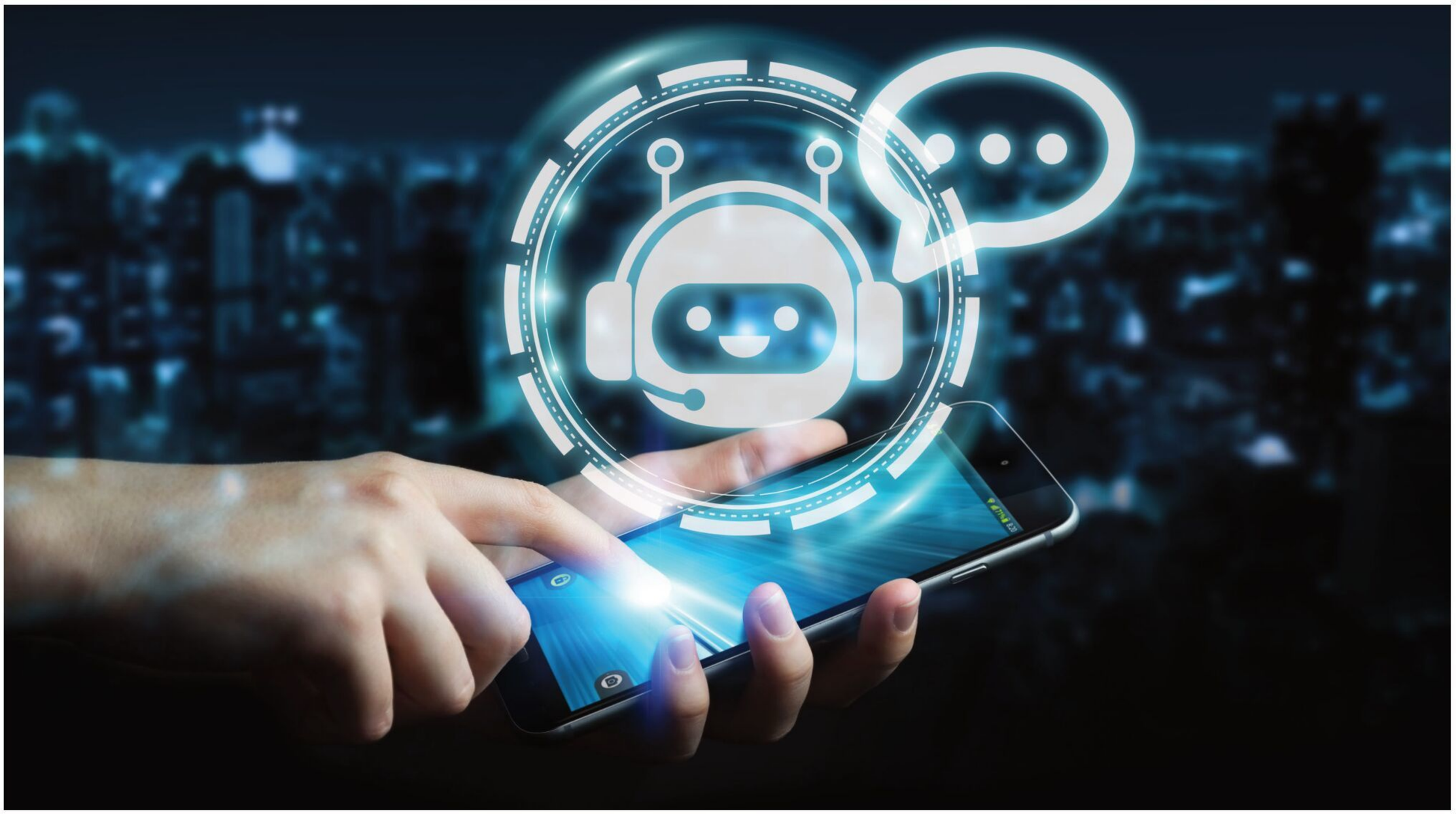
Argument	Description
xlab	label for x axis variable
ylab	label for y axis variable
title	plot title

You are encouraged to read the manual pages for the above R functions to learn more on their arguments, options and usage. END

By: Shakthi Kannan

The author is a free software enthusiast.

It's the Era of Conversational AI



Conversational AI helps customers interact with computer applications like chatbots just the way they would with humans. Let's explore this domain and take a look at what the tech giants are offering in this space.

Conversational artificial intelligence (AI) is today being used to implement various new age AI solutions like chatbots, virtual assistants, and contact centres, to name a few. Cloud based architectures like Azure AI, AWS ML or GCP ML provide many services suitable for building a chatbot combined with other native cloud services. AWS has even provided pre-build CloudFormation templates from Marketplace to swiftly develop a serverless chatbot service.

Smart devices like Amazon Echo and Apple HomePod use voice based intelligent virtual assistants (IVAs) to automate regular tasks. Amazon Echo uses Alexa, while Apple HomePod uses Siri. These simple software agents can take voice commands and execute tasks using an Internet connection and API interfaces.

Technical architecture of conversational AI chatbots

Typically, all IVA interfaces work using natural language processing (NLP) by segmenting audio inputs. For example, a command like "Siri, call Alan on his home number," will be split into each word using automatic speech recognition (ASR). The dictionary of phonetics will then be searched for a suitable mapped pattern to get the relevant action to execute the command by using an API interface. After executing the action, the response will be formulated using a text-to-speech (TTS) interface.

A chatbot is an IVA but is different from Siri/Alexa devices. A chatbot or conversational assistant is a dialogue based system that takes continuous inputs and uses previous chat messages to contextualise the response. Alexa/Siri are service agents that take commands and have an event driven approach, where a voice command is the event.

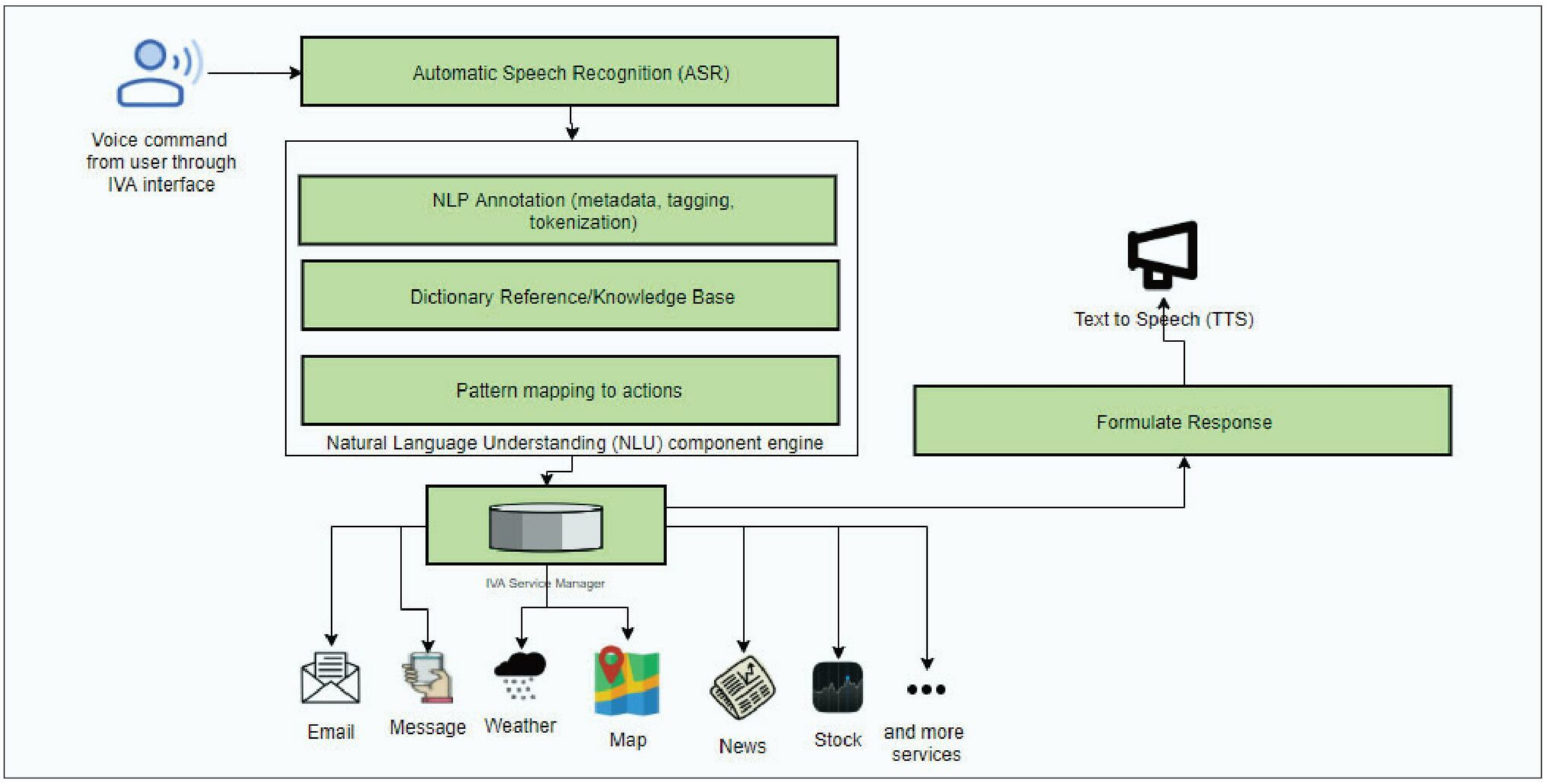


Figure 1: Execution steps of IVA

Chatbots vs conversational AI

Telemedicine and virtual health consultation are the new normal in the world after the recent pandemic. Hence, small clinics to large medical institutions prefer to develop and deploy a health bot, which can help patients with remote consultation. Health bots typically use AI and ML to process the query written by users through NLP, search for the response from their knowledge base, and have an interactive discussion with them. Hence, they are termed conversational AI enabled health bot solutions.

Chatbots and contact center AI play a critical role in developing a better customer experience. Traditionally, conversational AI was built by training the system to build the knowledge base and worked with a concrete set of functionalities only. With modern AI/ML services, self-managed conversational AI applications can be built very easily.

Azure Bot Service offers an AI agent that interacts with humans for support activities such as virtual banking assistance, insurance advice, IT helpdesk support and medical consultation, to name a few. The conversational AI solutions it offers are built on the following technical components.

Bot logic, which includes language understanding (LUIS) for predicting user intent from conversational natural language, QnA Maker to prepare questions and answers for FAQs and quick help, and Conversation Learner to automatically build the knowledge base, based on example interactions.

Speech capture converts speech to text using specific vocabulary and by understanding various styles of speech. Machine translation is used to translate text from different languages supported by the bot service.

Speech synthesis prepares the response from the knowledge base and pre-built answers, and uses a text to speech service to speak out responses from pre-built voices or create a custom voice. It helps users feel they are interacting with human assistants.

Some common use cases in conversational AI

Conversational AI is one of the most promising sectors of AI, with the global market for it projected to grow from US$ 4.2 billion in 2019 to US$ 15.7 billion by 2024. It is being used in both the commercial and domestic sectors today.

Here are a few use cases that highlight the revolution that conversational AI truly is.

Customer service: Conversational AI delivers the most impactful customer support services. It is now not only concerned with helping customers find timely solutions but is also making the life of agents better. Here are some examples.

- *Better agent experience:* AI helps agents do their jobs better; with the help of conversational AI, support executives can get through tickets much faster and have high quality interactions with customers.
- *More self-service:* By using a tool that can better leverage the knowledge base and company data, organisations can help customers help themselves better than ever before. AI can help answer the most frequently asked questions, giving customers what they need and deflecting redundant tasks from support agents.

- *Improved support metrics:* Increased rates of self-service and ticket deflection mean one thing: an improvement in the support team's goals and metrics. Conversational AI is what a support team needs to help increase customer satisfaction by reducing response times.

E-commerce: According to *Forbes*, more than 2 billion people made online purchases in 2020 and 2021, and e-retail sales totalled US$ 4.2 trillion globally. With such massive volumes, there is clearly a significant opportunity to utilise conversational AI for e-commerce businesses. Here are a few examples.

- *Automated purchasing:* Conversational AI can help customers automatically make purchases online, set up automated repeat orders, and deal with any issues regarding their purchases.
- *Insightful data:* AI isn't just helpful for the people using it; it can actually give very insightful data for businesses as a whole. By taking in data every single day, AI can help you learn about customer buying habits, get feedback, and help you strategise.
- *Boost customer satisfaction metrics like NPS and CSAT:* The best customer insights come after their interactions. With AI you can access data, analyse it using customer metrics and use it to improve business operations across the board.

Banking and financial services: Conversational AI is being used to reduce friction and make banking easy for customers to manage on their own. Besides automating functions and conversations, AI has helped reduce operating costs.

- *Account management:* Conversational AI or a banking virtual assistant can check customer balances and process transactions across all bank accounts, help make transfers, freeze accounts, or increase limits with the help of artificial intelligence.
- *Analysis of data:* Businesses can learn a lot from customers' banking habits, using the information to push initiatives and create goals that will improve customer experience.

Education: Education is one of the top industries benefiting from automated interactions and artificial intelligence. With Covid-19 moving learning online and institutions going through digital transformations to keep up with the times, it's clear that there is an opportunity to change things within education and leverage AI.

Conversational AI has shown that the education industry is on track to make learning more personalised, accessible, feasible, streamlined, and instant. This is what AI is doing for education.

- *Artificial teaching assistants:* AI can help with personalised learning by providing individual artificial teaching assistants that learn about the student and can adapt to their study habits with them.
- *Boost performance:* AI can help figure out where students can perform better.
- *Set learning pace:* Everyone learns at their own pace. If students are following an online learning course, AI can help set the pace for their personalised learning.

Retail: According to a recent IBM study,the adoption of AI in retail and consumer products is expected to grow another 40 per cent in the next three years.

Here are a few examples of conversational AI implementation in retail.

- *Searches:* AI can help you get the most up-to-date information regarding prices and deals. Machine learning algorithms now show you personalised information based on past retail searches.
- *Order management:* Intelligent chatbots have come a long way. They can now be used to manage shopping orders, tracking order status with a bot that can automatically find the information you need.
- *Account management:* Chatbots can also help with account management. There's no need for human interference when AI can reset passwords and set up two-factor authentication for customers.

Insurance: Insurance is finally undergoing a digital transformation and it's using AI to help it. Here are a few examples.

- *Assessing risk:* With the emergence of machine learning algorithms, it's become even easier for underwriters and insurance agents to bring in more information and data in order to gauge the risk being taken by insurance companies.
- *Fraud detection:*Insurance is complicated; it is not uncommon for people to lie on insurance applications to try to get better premiums or better coverage. AI can help detect fraud by having accuracy parameters set on applications in order to ensure that information being sent in is accurate and truthful.
- *Reducing human error:* The chain between a person and their insurance policy is wide and complex; there is a series of middlemen who examine information between the insured and the carrier. A lot of information can get lost due to human error and manual work. Conversational AI can reduce human error by analysing documents and data for you, minimising the friction and possibility for mistakes.

Travel: With conversational AI, the travel industry has reached new heights.

- *Booking:* AI can help travellers book their flights! Chatbots can be deployed to help travellers find the best deals and secure them.
- *Skychannel bots:* If a friend or loved one is travelling, you can scan the sky for their flight using AI tools that tell you exactly where the flight is currently.

Now let's briefly explore the conversational AI services being offered by the tech giants.

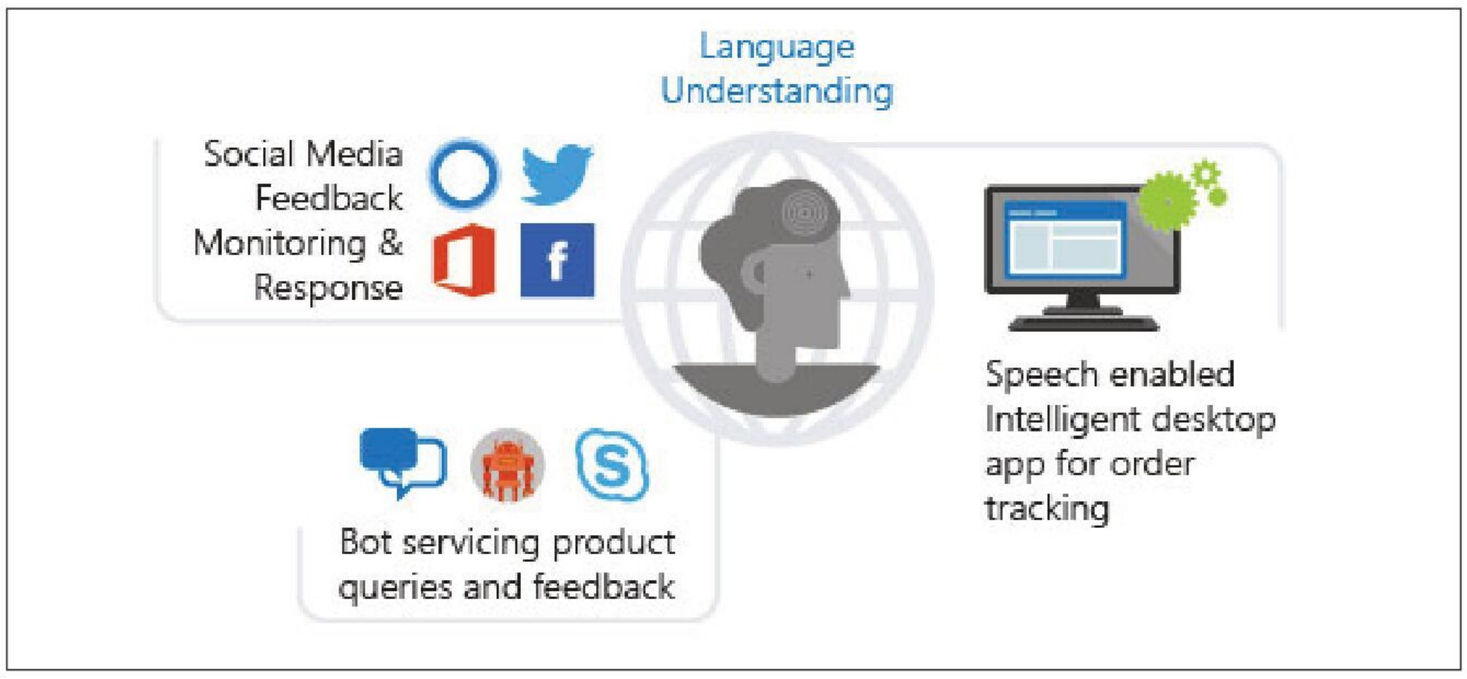


Figure 2: Azure LUIS and bot services (Image Source: Azure Documentation)

Azure Bot Service

Azure Language Understanding (LUIS) is a cloud API service from Microsoft, which uses custom ML services for conversational AI solutions like chatbot development. It uses an NLP technique called natural language understanding (NLU).

LUIS has three components – utterances, intent and entities. Utterances are plain text sentences which are thrown by users as a question to the bot. For example, "What time is the movie being shown in Casino?" Intents are the specific actions represented in the utterances. For example, 'show time' is the intent in the above example. Entity is the reference used in the utterance. 'Casino theatre' is the entity in this example.

LUIS can also be used for filtering negative or positive intents from social media posts.

When there is a user input as a sentence, LUIS takes the utterances and does NLU processing. It gets the intent score as a value between 0 and 1. Based on the best score fitment, it gets the entities associated and fetches the response for the same from the knowledge base.

LUIS can be used to create custom language processing capability for any local language by training the model to process new utterances of a custom language model. Also, there are built-in security features available to keep the LUIS API accessible in a secured way. LUIS can run on Azure cloud, on-premises or on the edge, as well as by installing LUIS in a Dockerized container.

Amazon Lex

Amazon Lex is a service for building conversational interfaces into any application, using voice and text. It provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognise the intent of the text, to help build applications with highly engaging user experiences and lifelike conversational interactions

Amazon Lex enables any developer to build conversational chatbots quickly. No deep learning expertise is necessary—to create a bot, you just specify the basic conversation flow in the Amazon Lex console. It manages the dialogue and dynamically adjusts the responses in the conversation. Using the console, you can build, test, and publish your text or voice chatbot. You can then add the conversational interfaces to bots on mobile devices, web applications, and chat platforms (for example, Facebook Messenger).

Amazon Lex provides pre-built integration with AWS Lambda, and you can easily integrate with many other services on the AWS platform, including Amazon Cognito, AWS Mobile Hub, Amazon CloudWatch, and Amazon DynamoDB. Integration with Lambda

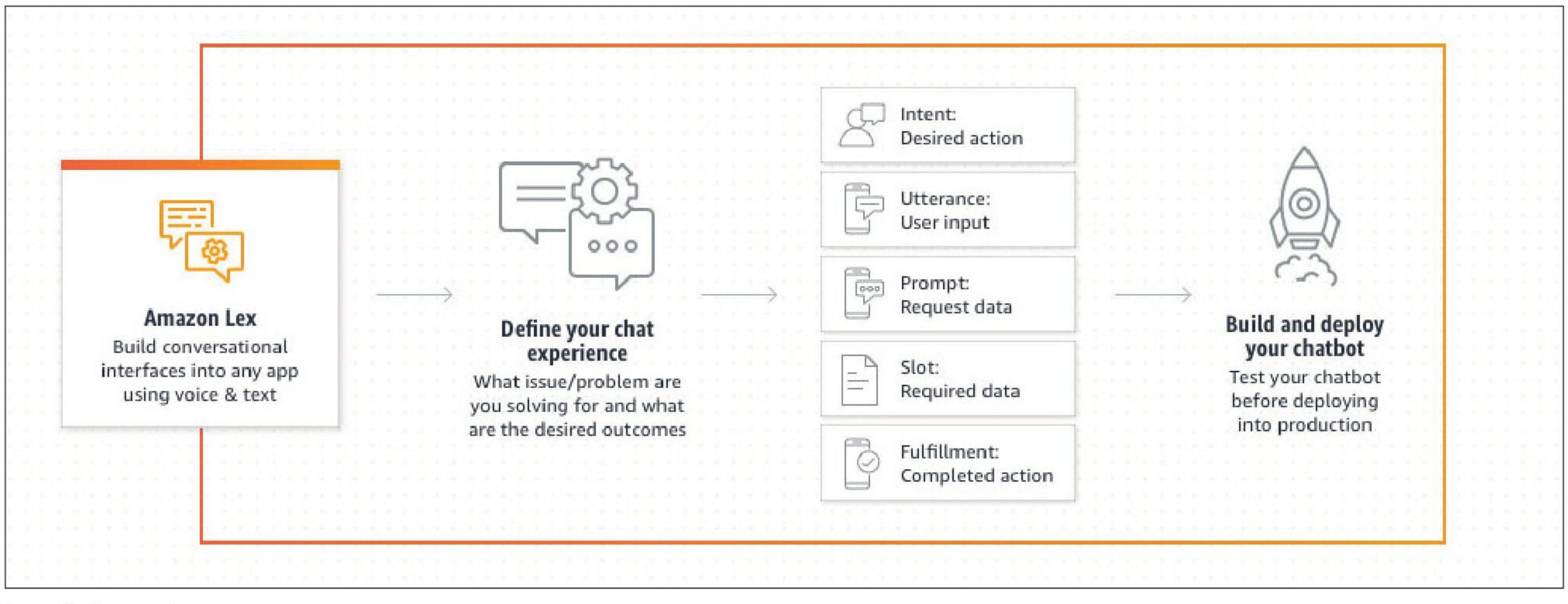


Figure 3: Amazon Lex

provides bots access to pre-built serverless enterprise connectors to link to data in SaaS applications, such as Salesforce, HubSpot, or Marketo.

The following are the benefits of Amazon Lex:

- Integrated role-based access control across all AWS services (IAM)
- Comprehensive, cross-service API audit logging and security (CloudTrail)
- Integration with other AWS services (24x7 support and consolidated billing)
- Training and architectural patterns/ guidance (well architected)
- A fully managed service, Amazon Lex scales automatically
- Easy-to-use console to guide you through the process
- Built-in integration with the AWS platform
- Amazon Lex bot processes voice or text input in conversation
- Helps build a complete natural language model

Google Contact Center AI

Google Cloud provides loads of AI based solutions, which are integrated with Google Contact Center AI services for virtual assistance.

A few important components that strengthen the capabilities of Contact Center AI (CCAI) of Google Cloud, and are in general availability mode (GA) now, are listed below.

Dialogflow CX: During customer assistance calls, complex queries need to be handled. Supplemental questions need a quick search of the knowledge base to quickly help human agents frame an appropriate reply. Dialogflow CX agents build on Google's NLU algorithms to handle these search capabilities.

Agent Assist: This is built with AI assisted conversational services to prepare recommended answer choices for human agents to choose from when replying to customer queries. It uses a centralised knowledge base to prepare ready-to-use responses and also transcribe live calls in different languages. This aids AI enabled processing in the backend for training AI models to analyse frequently asked questions, answer patterns and customer groups based on interest.

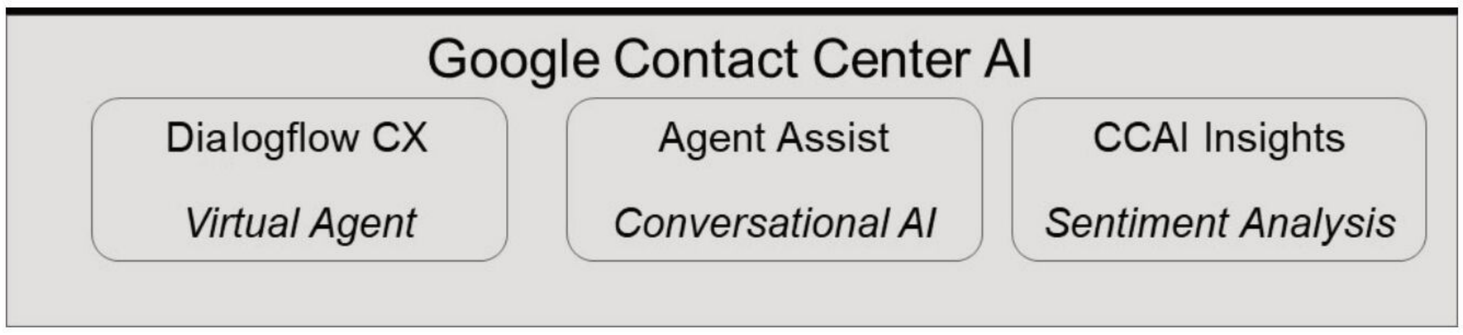


Figure 4: Google Contact Center AI

CCAI Insights: From voice enabled calls to chat based text analysis, CCAI Insights helps to prepare metrics analysis, sentiment analysis and ratings to answers using the NLP capabilities of Google Cloud. This helps to analyse customer interactions and improve customer connects (chat/voice).

Google CCAI services include text to speech and speech to text conversion, sentiment analysis, ranking of responses, integrated IVR services and NLP to provide unified customer support powered by Google Cloud AI services.

IBM Watson

IBM Watson is a data analytics processor that uses NLP. It performs analytics on the vast repositories of data that it processes to answer human-posed questions, often in a fraction of a second.

IBM Watson's cognitive and analytical capabilities enable it to respond to human speech, process vast stores of data, and return answers to questions that companies could never solve before.

The advantages of IBM Watson AI are:

- It gives you complete control of what is important to you and therefore the foundation of your competitive advantage, your data, models, learning, and API.
- Watson learns more from less because of its high learning power.
- It was initially available only on IBM Cloud but is now portable across any cloud-powered business. This prevents customers from being locked into one vendor and enables them to start deploying AI wherever their data resides.

With new AI solutions like Meta's BlenderBot and Microsoft Cortana, conversational AI has moved up to the next level in new age solutions across different sectors. END

References

- *https://www.microsoft.com/en-us/ai/ai-lab-conversational-ai*
- *https://healthtechmagazine.net/article/2021/07/what-microsofts-nuance-acquisition-means-healthcare-industry*
- *https://azure.microsoft.com/en-us/services/bot-services/health-bot/*
- *https://docs.microsoft.com/en-us/azure/cognitive-services/luis/what-is-luis*
- *https://aws.amazon.com/lex/*
- *https://cloud.google.com/solutions/contact-center*
- *https://www.ibm.com/watson*

By: Dr Anand Nayyar and Dr Magesh Kasthuri

Dr Anand Nayyar is a PhD in wireless sensor networks and swarms intelligence. He works at Duy Tan University, Vietnam, and loves to explore open source technologies, IoT, cloud computing, deep learning and cyber security.

Dr Magesh Kasthuri is a senior distinguished member of the technical staff and principal consultant at Wipro Ltd. This article expresses his views and not that of Wipro.

Testing SaltStack Environment with KitchenSalt, Docker and TestInfra

Have you ever struggled to test SaltStack states or formulas before rolling them out into production, or wondered how to validate SaltStack code changes without breaking the existing SaltStack environment? This tutorial shows how to design a seamless testing and validating workflow using KitchenSalt, Docker and TestInfra to test the SaltStack environment independently.



SaltStack or Salt is one of the most potent configuration management tools to automate infrastructure at scale. It facilitates users to efficiently manage the end-to-end life cycle of the servers, starting from bootstrapping a new node to deploying a live application on it. Despite being a robust automation engine, there is no easy way to test and validate the Salt code changes other than running them on individual servers and checking the results manually. Doesn't sound like an ideal practice, especially when dealing with a massive infrastructure and considering a bunch of developers working simultaneously on the same Salt code. Moreover, making such changes in production without independent testing could lead to a non-functioning Salt infrastructure.

But KitchenSalt and TestInfra reduce that risk. They allow users to replicate an independent infrastructure that is similar to the production environment locally and execute end-to-end validations against it.

What follows in this article assumes that you have the following knowledge:

- Understanding of SaltStack and its basic building blocks
- Understanding of Docker and its core commands
- Basics of Python & Ruby (Gem) for installing dependent libraries

You will need the following in order to design the workflow:

- Latest and stable version of Docker
- Ruby version >=2.6, along with Test-Kitchen, Kitchen-Docker and Kitchen-Salt libraries
- Python version 3.x, along with PyTest and TestInfra modules

```
[root@ip-172-31-31-154 kitchen-salt]# docker --version
Docker version 1.13.1, build 7d71120/1.13.1
[root@ip-172-31-31-154 kitchen-salt]# ruby --version
ruby 2.7.6p219 (2022-04-12 revision c9c2245c0a) [x86_64-linux]
[root@ip-172-31-31-154 kitchen-salt]#gem --version
3.0.3.1
[root@ip-172-31-31-154 kitchen-salt]#bundle --version
Bundler version 2.3.24
[root@ip-172-31-31-154 kitchen-salt]# python3 --version
Python 3.6.8
[root@ip-172-31-31-154 kitchen-salt]# pip3.6 install -r requirements.txt
..................................................
Successfully installed attrs-22.1.0 certifi-2022.9.24 charset-normalizer-2.0.12 idna-3.4 importlib-metadata-4.8.3 iniconfig-1.1.1 packaging-21.3 pluggy-1.0.0 py-1.11.0 pyparsing-3.0.9 pytest-7.0.1 pytest-testinfra-6.8.0 requests-2.27.1 testinfra-6.0.0 tomli-1.2.3 typing-extensions-4.1.1 urllib3-1.26.12 zipp-3.6.0
```

How does it work?

To design a seamless testing workflow, we will use KitchenSalt in conjunction with TestInfra to improve the quality of states and reduce errors while deploying Salt code in a production environment.

KitchenSalt is built on top of Test Kitchen, which was initially developed to test and validate Chef workflows. It is a provisioner for SaltStack, which permits users to leverage Test Kitchen to validate the SaltStack environment locally without a Salt master or minions. On the other hand, TestInfra is a Python testing module written on top of the popular Pytest testing framework. With the help of TestInfra, users can run integration or unit tests to validate the actual state of the servers configured by SaltStack.

Figure 1 illustrates the logical representation of the entire workflow, where we will spin up a Docker container using KitchenSalt and apply Salt states or formulas to it. Once the provisioning is completed, a few integration tests will be run with the help of TestInfra to validate the state of the provisioned container.

Prepare Webserver formula

For this demonstration, we will need a working Salt formula. So, let's create a basic one to install and configure a Webserver instance. The formula consists of the following three essential states.

- *install.sls:* To install Webserver (Apache) packages on the instance
- *configure.sls:* To set up Webserver conf and *index.html* page
- *service.sls:* To start Webserver service once setup is completed

```
[root@ip-172-31-31-154 kitchen-salt]# cat webserver/install.sls
{% from "webserver/map.jinja" import vars with context %}
{{ sls }}.pkg:
  pkg.installed:
    - name: {{ vars['package'] }}
{{ sls }}.group:
  group.present:
    - name: {{ pillar['app_user'] }}
    - gid: {{ pillar['gid'] }}
    - require:
      - pkg: {{ sls }}.pkg
{{ sls }}.user:
  user.present:
    - name: {{ pillar['app_user'] }}
    - uid: {{ pillar['uid'] }}
    - gid: {{ pillar['gid'] }}
    - home: {{ pillar['doc_root'] }}
    - shell: /bin/nologin
    - allow_uid_change: true
```

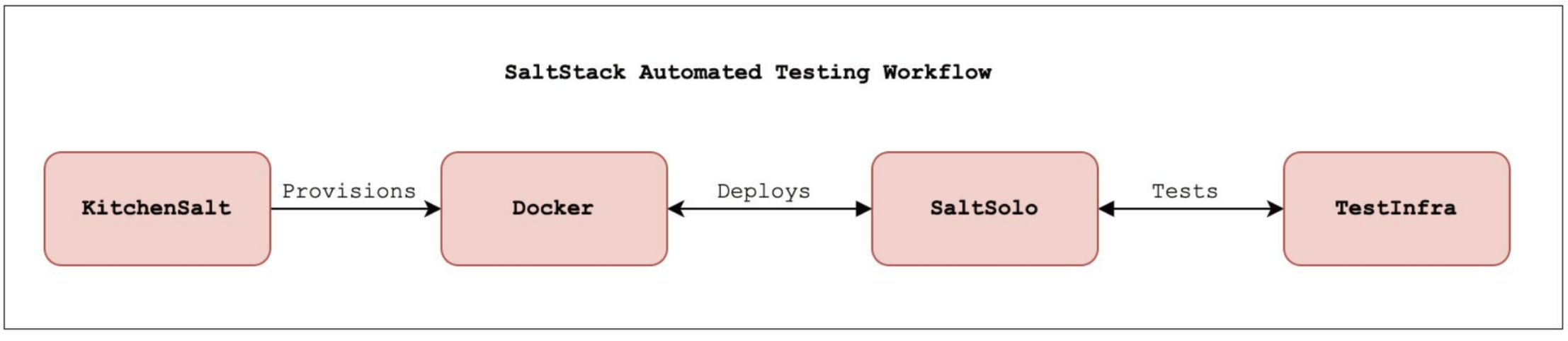


Figure 1: Logical representation of the entire workflow

```
    - require:
      - group: {{ sls }}.group
```

Now let us set up the webserver configuration file *config.sls* as below:

```
[root@ip-172-31-31-154 kitchen-salt]# cat webserver/config.sls
{% from "webserver/map.jinja" import vars with context %}
include:
  - webserver.install
{{ sls }}.index.html:
  file.managed:
    - name: {{ pillar['doc_root'] }}/index.html
    - source: salt://webserver/files/index.html
    - require:
      - sls: webserver.install
{{ sls }}.server.conf:
 file.managed:
    - name: {{ pillar['server_conf'] }}
    - source: salt://webserver/templates/httpd.conf.j2
    - template: jinja
    - require:
      - sls: webserver.install
```

Next, configure the *service.sls* to start webserver service once setup is completed.

```
[root@ip-172-31-31-154 kitchen-salt]# cat webserver/service.
sls
{% from "webserver/map.jinja" import vars with context %}
include:
  - webserver.install
  - webserver.config
{{ sls }}.service:
  service.running:
    - name: {{ vars['service'] }}
    - watch:
      - sls: webserver.install
      - sls: webserver.config
```

Installing Kitchen

Once the formula is ready, let's install the Kitchen-specific dependencies using Gemfile, which requires *kitchen-salt*, *kitchen-docker*, and *kitchen-sync* libraries.

```
[root@ip-172-31-31-154 kitchen-salt]# cat Gemfile
#Gemfile
source 'https://rubygems.org'
gem 'kitchen-salt'
gem 'kitchen-docker'
gem 'kitchen-sync'
```

Next, run the installer which installs all the required dependencies using Gem bundler:

```
[root@ip-172-31-31-154 kitchen-salt]# gem install bundler
Fetching bundler-0.0.1.gem
Successfully installed bundler-0.0.1
Parsing documentation for bundler-0.0.1
Installing ri documentation for bundler-0.0.1
Done installing documentation for bundler after 0 seconds
1 gem installed
[root@ip-172-31-31-154 kitchen-salt]# bundler install | grep
kitchen
Using test-kitchen 3.3.2
Using kitchen-docker 2.13.0
Using kitchen-salt 0.7.2
Using kitchen-sync 2.2.1
```

Configuring Kitchen

Once the Kitchen dependencies are sorted, let's create a *.kitchen.yml* file consisting of Test Kitchen configurations. This is the main file responsible for spinning up a Docker container and deploying the Salt formula.

The file includes the following major components.

- *driver:* Defines which driver needs to be configured and how to control it. In our example, we will use Docker as a default driver.
- *platforms:* Defines the platforms or distributions against which it tests Salt states or formulas. We will use CentOS7 as a default platform for this demonstration.
- *provisioner:* Defines what plugins to use to converge the system and how to apply the changes. The provisioner we use is salt-solo.
- *suites:* Defines the software suite to be used to test Kitchen against. In this context, Kitchen will test against 3004.2 release of Salt.
- *verifier:* Defines what tests need to be performed against the newly provisioned infrastructure. We will use TestInfra as a default testing framework to run our tests.

The next step is to put all the configs together:

```
[root@ip-172-31-31-154 kitchen-salt]# cat .kitchen.yml
driver:
  name: docker
  use_sudo: false
  privileged: true
  run_command: /usr/sbin/init
  forward:
    - 80
provisioner:
  name: salt_solo
  salt_install: bootstrap
```

```
salt_version: latest
salt_bootstrap_url: https://bootstrap.saltstack.com
is_file_root: true
require_chef: false
salt_copy_filter:
  - .git
  - .kitchen
  - Gemfile.lock
  - pytests_cache
  state_top:
    base:
      "*":
        - webserver
  pillars_from_directories:
    - source: pillars/
      dest: srv/pillar/
  pillars:
    top.sls:
      base:
        '*':
          - webserver
platforms:
  - name: centos-7
  - name: ubuntu
suites:
  - name: webserver
    provisioner:
      salt_bootstrap_options: -qX -p git -p curl -p sudo git
3004.2
verifier:
  name: shell
  remote_exec: false
  command: python3 -m pytest -v tests/
```

Using Kitchen

Now that we have learnt how to set up Test Kitchen, let's start testing the webserver formula and verify the same using Kitchen. Here are some helpful Kitchen commands that we will be using in this context.

- *List:* To show the current state of each configured environment
- *Login:* To log into the built environment
- *Converge:* To run the provision command, in this case salt_solo, and the specified states
- *Verify:* To run the verifier
- *Destroy:* To remove the created environment
- *Test:* Consists of creating, converging, verifying, and then destroying states mentioned above

With all this in place, we can now run Kitchen and see it in action.

1. First, get a list of the available environments configured in *kitchen. yml:*

```
[root@ip-172-31-31-154 kitchen-salt]# bundle exec kitchen list
Instance             Driver   Provisioner  Verifier  Transport
Last Action    Last Error
webserver-centos-7   Docker   SaltSolo     Shell     Ssh
<Not Created>  <None>
webserver-ubuntu     Docker   SaltSolo     Shell     Ssh
<Not Created>  <None>
```

2. Next, run the *converge* command, which launches a new CentOS7 container, connects to it, and then deploys the webserver formula on top of it.

```
[root@ip-172-31-31-154 kitchen-salt]# bundle exec kitchen
converge webserver-centos-7
-----> Starting Test Kitchen (v3.3.2)
-----> Creating <webserver-centos-7>.....
.....................................
Summary for local
      ------------
      Succeeded: 6 (changed=6)
      Failed:    0
      ------------
      Total states run:     6
      Total run time:  14.220 s
      Downloading files from <webserver-centos-7>
      Finished converging <webserver-centos-7> (2m3.93s).
-----> Test Kitchen is finished. (2m7.80s)
```

3. Then, run some tests against the newly built environment to ensure everything is as it should be:

```
[root@ip-172-31-31-154 kitchen-salt]# bundle exec kitchen
verify webserver-centos-7
-----> Starting Test Kitchen (v3.3.2)
-----> Verifying <webserver-centos-7>...
       [Shell] Verify on instance webserver-centos-7 with
state={:username=>"kitchen", :ssh_key=>"/root/kitchen-salt/.
kitchen/docker_id_rsa", :image_id=>"2879404546a1", :container_
id=>"db77fdc81097f502e38ad830cfa61e30d1bd0629df44c550435f0
a4ee87e2024", :hostname=>"localhost", :port=>32779, :last_
action=>"setup", :last_error=>"Errno::ENOENT"}
============================= test session starts
==============================
platform linux -- Python 3.6.8, pytest-7.0.1, pluggy-1.0.0 --
/bin/python3
cachedir: .pytest_cache
rootdir: /root/kitchen-salt
plugins: testinfra-6.8.0, testinfra-6.0.0
collecting ... collected 4 items
tests/integration/test_webserver.py::test_apache_service
PASSED          [ 25%]
tests/integration/test_webserver.py::test_index_file PASSED
```

```
[ 50%]
tests/integration/test_webserver.py::test_website_200 PASSED
[ 75%]
tests/integration/test_webserver.py::test_website_content
PASSED        [100%]
============================== 4 passed in 0.66s
==============================
/usr/local/lib/python3.6/site-packages/_testinfra_renamed.
py:10: DeprecationWarning: testinfra package has been renamed
to pytest-testinfra. Please `pip install pytest-testinfra` and
`pip uninstall testinfra` and update your package requirements
to avoid this message
  ), DeprecationWarning)
      Finished verifying <webserver-centos-7> (0m1.08s).
-----> Test Kitchen is finished. (0m1.27s)
```

4. Finally, destroy the entire setup if everything looks good:

```
[root@ip-172-31-31-154 kitchen-salt]# bundle exec kitchen
destroy webserver-centos-7
-----> Starting Test Kitchen (v3.3.2)
-----> Destroying <webserver-centos-7>...
       [Docker] Destroying Docker container db77fdc81097f502e
38ad830cfa61e30d1bd0629df44c550435f0a4ee87e2024
UID      PID    PPID    C       STIME    TTY    TIME       CMD
root    1207    32513     0     14:13      ?
00:00:00           /usr/sbin/httpd -DFOREGROUND
       1001          1208        1207               0
14:13              ?        00:00:00           /usr/sbin/
httpd -DFOREGROUND
       1001      1209       1207               0
14:13     ?     00:00:00    /usr/sbin/httpd -DFOREGROUND
       1001     1210     1207    0          14:13
?    00:00:00           /usr/sbin/httpd -DFOREGROUND
 1001   1211    1207   0         14:13              ?
00:00:00  /usr/sbin/httpd -DFOREGROUND        1001
1213               1207     0     14:13   ?    00:00:00     /
usr/sbin/httpd -DFOREGROUND        root              32513
32494              0         14:11             ?
00:00:00   /usr/sbin/init
  root      32556      32513     0    14:11        ?
00:00:00    /usr/lib/systemd/systemd-journald
root    32571      32513     0       14:11     ?
00:00:00           /usr/lib/systemd/systemd-udevd
root   32618     32513     0       14:11     ?
00:00:00           /usr/lib/systemd/systemd-logind
 dbus               32620              32513
0       14:11       ?             00:00:00           /
usr/bin/dbus-daemon --system --address=systemd: --nofork
--nopidfile --systemd-activation
 root               32656              32513
0  14:11      ?       00:00:00           /usr/sbin/sshd -D
root        32685     32513      0                  14:11
tty1   00:00:00           /sbin/agetty --noclear tty1 linux
 db77fdc81097f502e38ad830cfa61e30d1bd
0629df44c550435f0a4ee87e2024
  db77fdc81097f502e38ad830cfa61e30d1bd0629df
44c550435f0a4ee87e2024
     Finished destroying <webserver-centos-7> (0m0.99s).
-----> Test Kitchen is finished. (0m1.17s)
```

This article has explored KitchenSalt and TestInfra at a surface level; however, it can be extended further by implementing a CI/CD pipeline in Jenkins to test Salt code changes before pushing them into production directly. Another example could be testing the Salt states or formulas against future OS releases of distributions before making a major upgrade in SaltStack. Or validate them against multiple versions of SaltStack to ensure there are no significant regressions. END

References

- *https://github.com/vijayg92/kitchen-salt*
- *https://testinfra.readthedocs.io/en/latest/*
- *https://github.com/saltstack/kitchen-salt/blob/master/docs/gettingstarted.md*
- *https://docs.saltproject.io/en/latest/topics/development/conventions/formulas.html*

By: Vijay Gosai

The author is an open source enthusiast and is currently working as a senior site reliability engineer. His interest areas include distributed systems, cloud computing, infrastructure automation, software development, design patterns, DevOps and emerging technologies.

THE COMPLETE MAGAZINE ON OPEN SOURCE

OpenSource ForYou

The latest from the Open Source world is here.

OpenSourceForU.com

Join the community at facebook.com/opensourceforu

Follow us on Twitter @OpenSourceForU

Setting Up an Email Server with Carbonio CE

This article gives an overview of how to deploy and manage a private email server on Linux using Carbonio Community Edition and its dedicated Admin Panel.



Sending and receiving confidential information is a normal part of any business. Many companies send this information through public email servers, which means they have little control over who can run, use and manage the email system. This exposes them to spam, phishing, and data breaches. Having a private server is a big step towards email security; plus, it keeps those creepy online ads away. Here is an easy way to deploy and manage your own private email server on Linux using Carbonio Community Edition and its dedicated Admin Panel.

What is Carbonio Community Edition?

Carbonio CE is a relatively new open source email server and collaboration platform by Zextras. It comes with an all-inclusive web client that contains a variety of tools for the common workplace tasks, such as files, calendars, chats, and video meetings; plus, a dedicated anti-spam and anti-virus feature.

Why Carbonio Community Edition?

Carbonio CE is free, open source, and multi-platform. It can boost teamwork and save time that is often wasted because of inefficient communication between team members. It is also available for mobile devices to facilitate communication even further, being accessible anywhere and at any time.

Carbonio CE focuses on enabling data sovereignty for enterprises by utilising self-hosted servers on the company's premises.

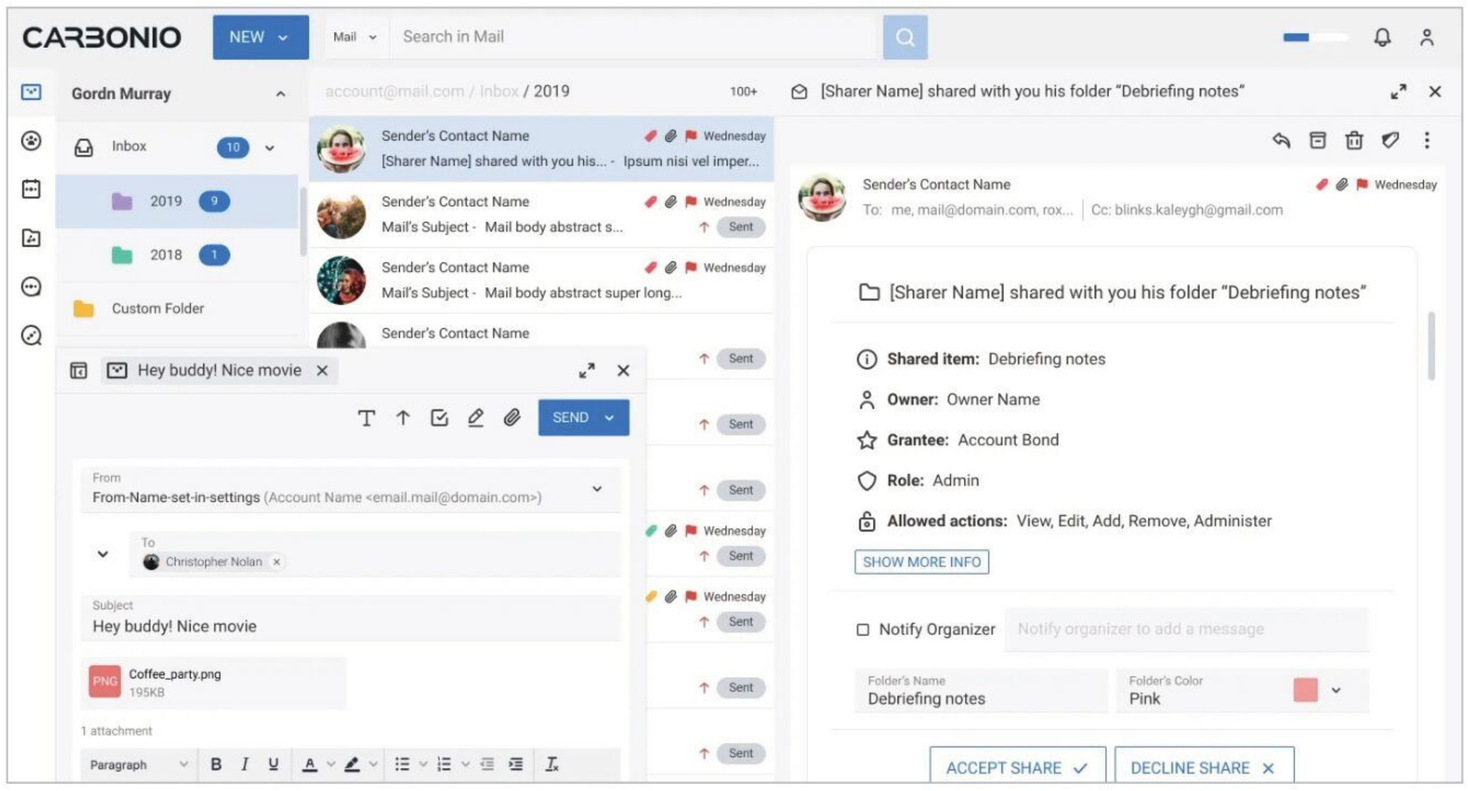


Figure 1: Carbonio CE web client

Quick installation

Installation of Carbonio CE is very straightforward. There are some requirements that should be met beforehand including:

- 1.5GHz 64-bit CPU
- 8GB of RAM
- 40GB of disk space
- FQDN (fully qualified domain name) such as *https://mail.example.com*
- Valid A and MX records for the server and domain
- Ubuntu 20.04 LTS Focal Fossa

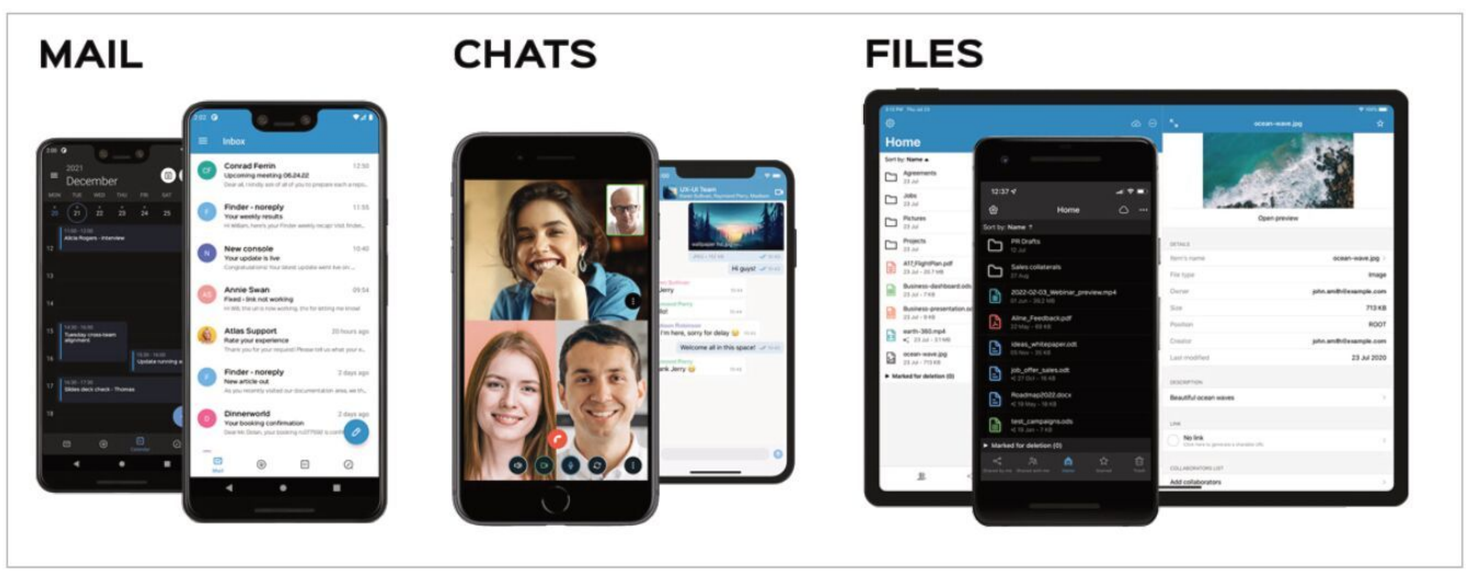


Figure 2: Carbonio CE in action

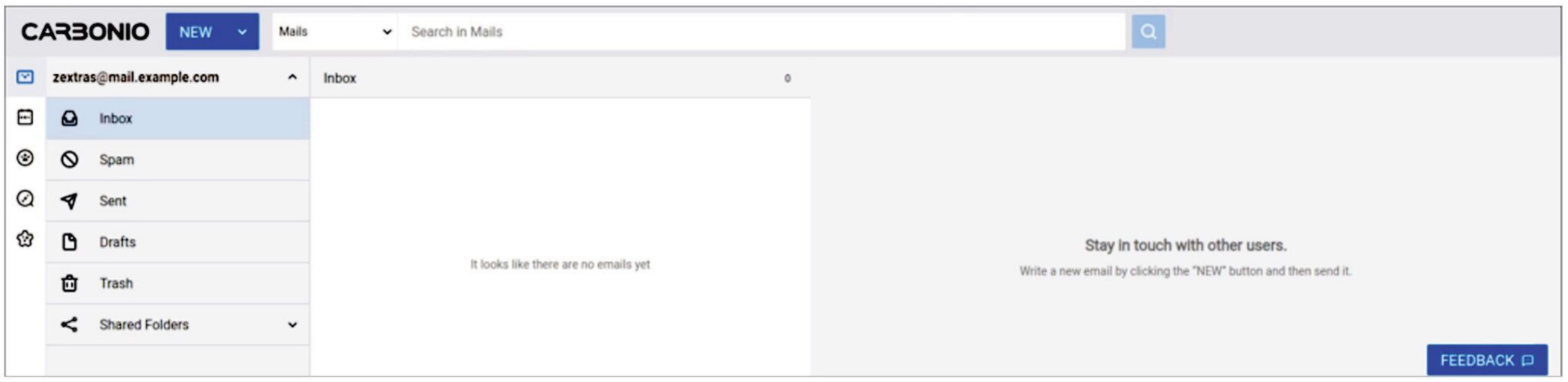


Figure 3: Carbonio CE inbox

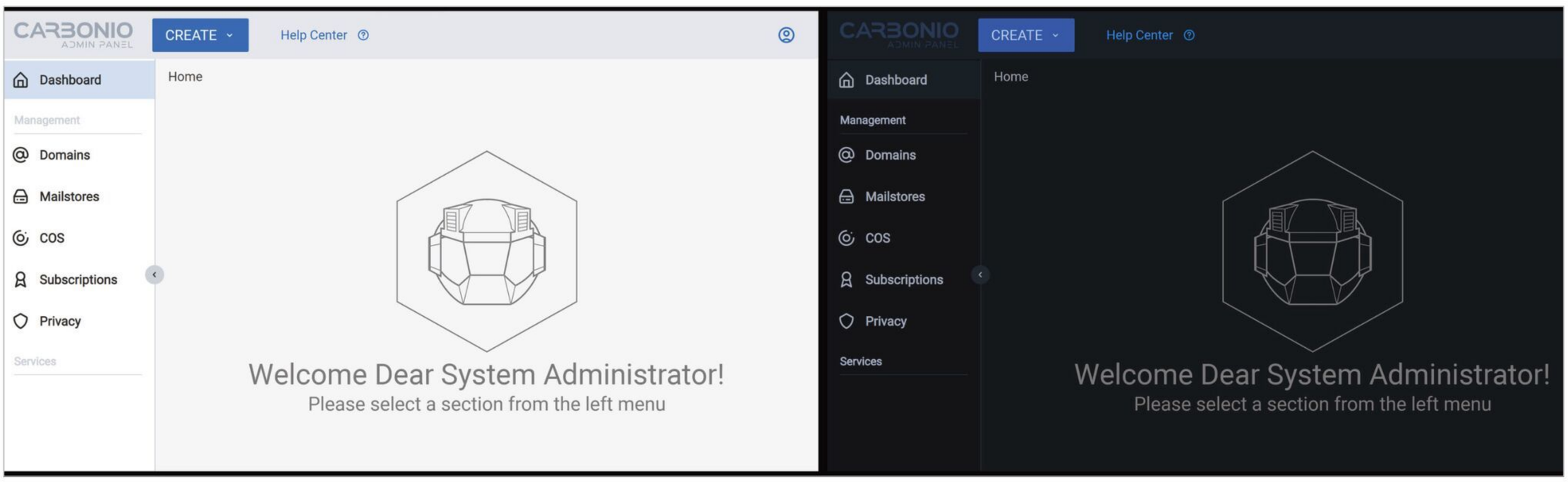


Figure 4: Admin Panel

Fill out the form by going to the Carbonio Community Edition web page. An email listing simple steps for adding the repository to your operating system will shortly be sent to you. After adding the repository successfully, just follow the on-screen instructions.

Add your local IP address and host name to your *hosts* file:

```
echo “$LOCAL_IP $HOSTNAME.$DOMAIN” >> /
etc/hosts
```

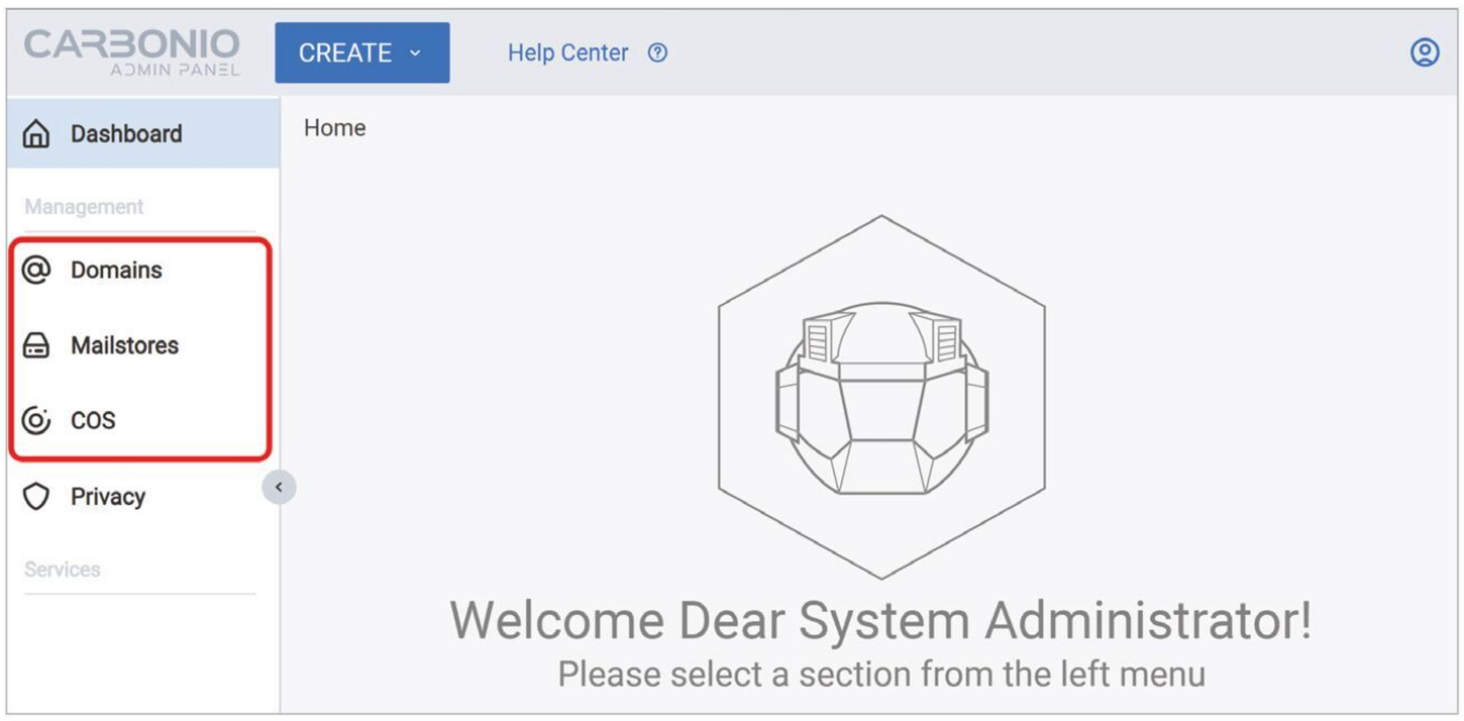


Figure 5: Carbonio CE server configuration menu

Install some necessary packages as the root user:

```
apt install service-discover-server \
carbonio-directory-server \
carbonio-proxy \
carbonio-webui carbonio-files-ui \
carbonio-admin-login-ui \
carbonio-mta \
carbonio-appserver carbonio-logger \
carbonio-user-management \
carbonio-files-ce carbonio-files-db \
carbonio-storages-ce \
carbonio-preview-ce \
carbonio-docs-connector-ce carbonio-
docs-editor \
carbonio-admin-ui carbonio-admin-
console-ui \
postgresql-12
```

You will configure your server using *carbonio-bootstrap* in this phase:

```
screen carbonio-bootstrap
```

Congratulations! You’re done and your private Carbonio CE server has been successfully deployed.

Users can now access the web client via the IP address or the domain such as:

- *https://mail.example.com/*
- *https://yourIP/*

If you have any problem during the installation you can read the official documentation for more information.

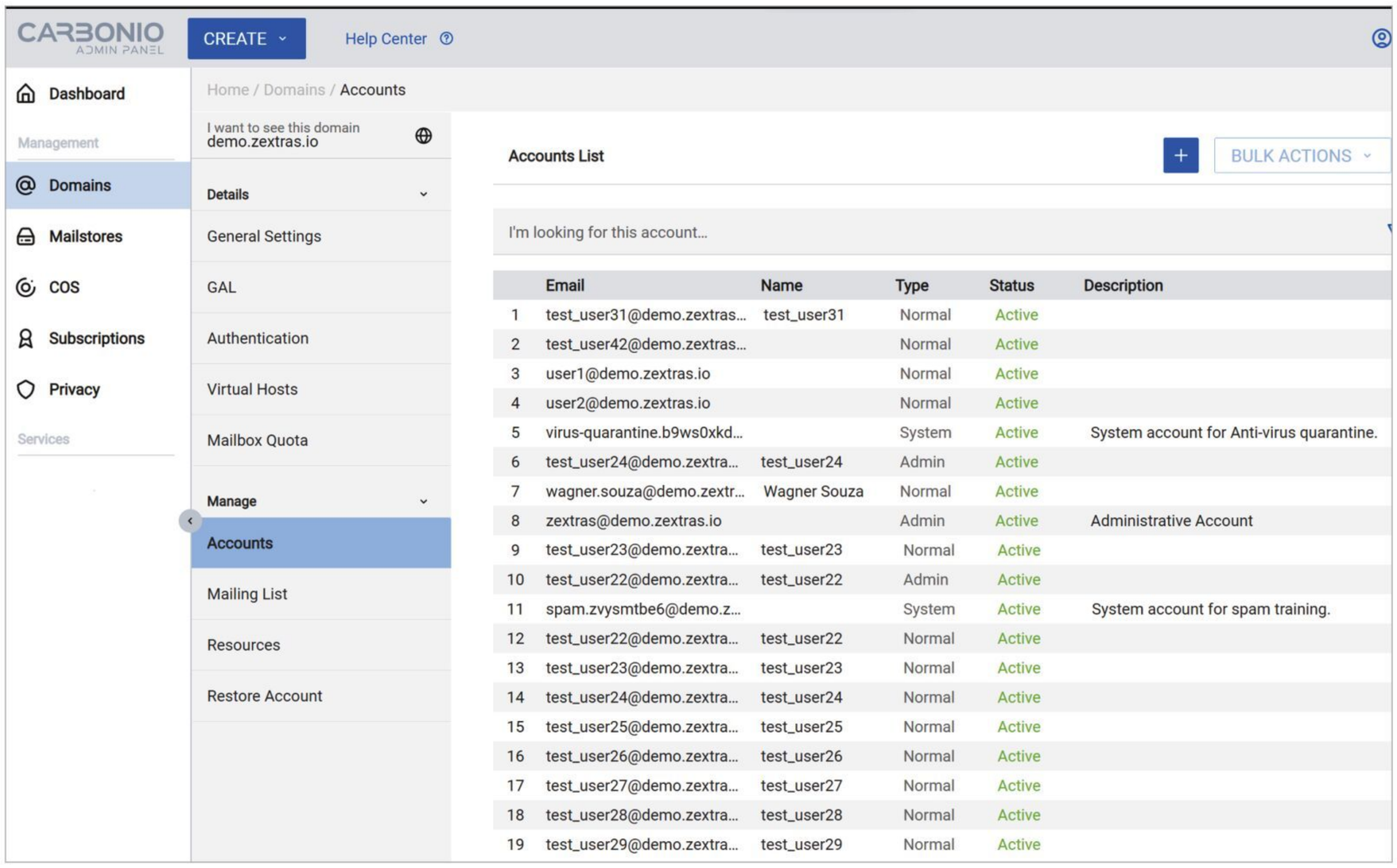


Figure 6: Creating user accounts

Bonus feature: Carbonio CE Admin Panel

As said before, Carbonio CE focuses on data sovereignty more than anything else. To do that, it uses self-hosted servers on the premises of the company. On-premises servers require more attention from systems administrators, which demonstrates the significance of an administrative console in making sysadmins' tasks easier.

In October 2022, Zextras launched a centralised and graphical interface for managing administrative operations, that is, the Admin Panel for Carbonio Community Edition.

Sysadmin-specific software evolves like all software, but at a much slower pace. Carbonio CE Admin Panel can be thought of as an evolution to classical admin tools, but more efficient. It has the important advantage of providing an overview of the whole system, all in one place. Monitoring and managing are all bundled into the panel, so you can view and make changes based on what you see. It can also be accessed remotely; after all, having gone through Covid we have all realised how important remote access is.

Let's be clear here though: what you can do on the Carbonio CE Admin Panel can also be done using the CLI, but it will in no way be as plain and intuitive. Plus, you will be able to view real-time analytics on your system and save time as every action will need only a simple click.

Overview of Carbonio CE Admin Panel

Carbonio CE Admin Panel offers a condensed, practical, and understandable interface that aims to provide the most convenient management of domains, users, and classes of service in Carbonio CE. It is available on port 6071 of the server's domain name. For instance, *https://mail.example.com:6071/login*.

Admin Panel follows the same design system used in Carbonio CE, and is available both in light and dark modes.

The primary bar on the side of the UI includes the essential categories to configure a Carbonio CE server, which are:

- *Domain management:* Tools for creating, editing, and removing domains are available in this area. Additionally, there are options for adding, changing and removing user accounts.
- *Mailstore management:* Information on the mailstores and servers that are part of the infrastructure is given in this section, along with the configuration tools used to set them up.
- *Class of service management:* Tools for configuring classes of service on the server are available in this area for creation, deletion, and modification.

Carbonio CE Admin Panel tries to simplify the most frequent tasks done by system administrators on email servers. For instance, managing accounts is one of a system administrator's most common activities. This is only a few clicks away using the Admin Panel — *Domain > Select Domain > Accounts*.

There you have it! You can now enjoy your private email and collaboration platform, and manage it easily through its dedicated admin panel. END

References

- *https://www.zextras.com/carbonio-community-edition/#discoverproduct*
- *https://docs.zextras.com/carbonio-ce/html/single-server-installation.html*

By: Arman Khosravi

The author is a technical writer at Zextras.

Connecting Arduino to the Internet

If you have just begun working with Arduino, connecting to the internet can be a very daunting task. I remember my own struggles with it in my early days of Arduino coding. With the lack of a proper interface and no inbuilt debugging tools, it can be hard to figure out where you went wrong. In this article, we shall cover everything you need to know about connecting your Arduino board to the internet, including the code.

When I first began playing around with my Arduino Uno, I was very proud of making LEDs blink and sensors beep. But I soon realised that the true potential of an Arduino can only be achieved once you connect it to the internet.

Pretty much any intermediate to advanced level Arduino project will require you to connect to the internet for various reasons, be it logging data on the cloud such as weather data or passing on real-time commands to your device remotely such as via an app on your phone.

In this tutorial article, we are going to look at how to connect your Arduino device to the internet comprehensively. Everything — from the hardware to the circuit and code — will be covered.

Connecting via the Ethernet

The first option for connecting your Arduino to the internet is via an Ethernet cable. If you are using an Arduino board that comes with a built-in Ethernet port such as an Arduino Yún, then you can skip the 'Hardware requirements' section and the circuit design description given below. Just connect the Ethernet cable to your device and start coding. However, if, like most people, you own a simpler version of Arduino such as Arduino Uno that does not have a built in Ethernet port, then you will have to buy a separate device called 'Ethernet Shield' to attach to your Arduino.

Hardware requirements

You will require an Ethernet cable with internet, an Arduino Uno board and an Ethernet Shield. Here's a description of the circuit design.

1. Connect the Ethernet Shield on top of the Arduino Uno.
2. Connect the Ethernet cable to the Ethernet Shield.

Figure 1 shows the circuit design of Arduino Uno and the Ethernet Shield. After connecting the two, your Arduino Uno should look like the one in Figure 2.

The code

Before getting started with the code for connecting the Arduino to the internet, we first need to include an external library to use in the program. This library will help us establish a connection and send/receive data over the internet.

Note: This library should come preinstalled with your IDE. If for some reason you run into errors, please download this library from the official GitHub repository of the Arduino Project.

```
#include <Ethernet.h>
```

In the next step, we will be defining some constants and variables required

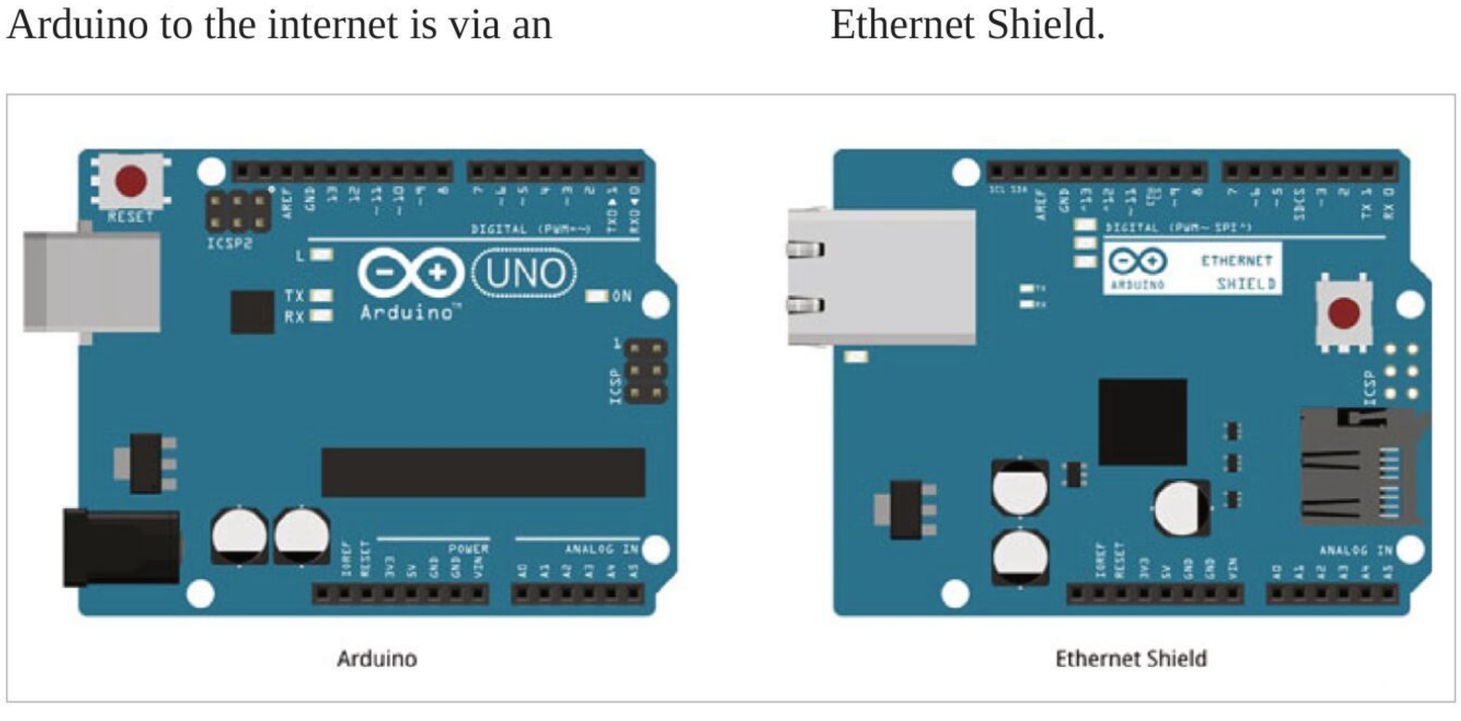


Figure 1: Arduino Uno and Ethernet Shield

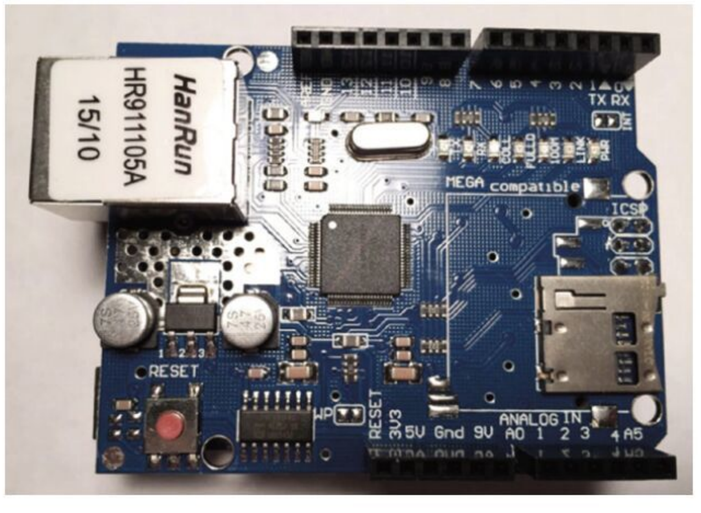


Figure 2: Ethernet Shield attached to Arduino Uno

for this program. First, we need the MAC address. This is often printed on a sticker on your Ethernet Shield. Next, we need to define a static IP address. Make sure that the IP address that you are adding isn't being used by any other user. Lastly, we shall define the *EthernetClient* variable.

```
byte mac[] = { oxDE, oxAD, oxBE, oxEF,
oxFE, oxED };
IPAddress staticIP(10, 0, 0, 20);

EthernetClient client;
```

We can now write the method for connecting to the internet. There are two steps here. First, we try to get an IP via the DHCP (dynamic host configuration protocol), i.e., try to fetch your dynamic IP. If that step fails, then we fall back to the *staticIP* that we have defined above.

```
void connectToInternet()
{
     // Step 1 - Try connecting with
DHCP
     If (Ethernet.begin(mac) == 0)
     {
       Serial.print("[ERROR] Failed to
connect via DHCP");
       Ethernet.begin(mac, staticIP);
// Connect via static IP defined earlier
}
// Add a delay for initialization
delay(1000);

Serial.println("[INFO] Connection
Successful");
Serial.print("");
printConnectionInformation(); // Custom
method
Serial.print
ln("---------------------------------");
Serial.println("");
}
```

As you can see in the code above, we used a custom method *printConnectionInformation()* for displaying the connection information.

So let us go ahead and write the code for that.

```
void printConnectionInformation()
{
     // Print IP Address
     Serial.print("[INFO] IP Address:
");
     Serial.println(Ethernet.
localIP());

     // Print Subnet Mask
Serial.print("[INFO] Subnet Mask: ");
     Serial.println(Ethernet.
subnetMask());

     // Print Gateway
     Serial.print("[INFO] Gateway: ");
     Serial.println(Ethernet.
gatewayIP());

     // Print DNS
     Serial.print("[INFO] DNS: ");
     Serial.println(Ethernet.
dnsServerIP());
}
```

Finally, we shall write the standard functions for the program, i.e., *setup()* and *loop()*.

```
void setup()
{
     Serial.begin(9600);

     // Connect to the internet
     connectToInternet();
}

void loop()
{
     // Nothing much to do here.
}
```

If everything checks out, then you should be seeing an output similar to Figure 3 on the *Serial Monitor* window.

Connecting via Wi-Fi

The second option is to connect your Arduino to the internet wirelessly via Wi-Fi. If, like most people, you own an Arduino Uno or any other Arduino board that does not have a built-in Wi-Fi capability, then you will have to buy a 'Wireless Shield' separately, just as we bought the Ethernet Shield in the previous section.

If you own an Arduino Yún or any other Arduino board with built-in wireless capability, then you can skip the 'Hardware requirements' and the description of the circuit design given below and get started with the code right away.

Hardware requirements

You will need an Arduino Uno and a Wireless Shield. Here's a description of the circuit design.

1. Connect the Wireless Shield on top of the Arduino Uno.

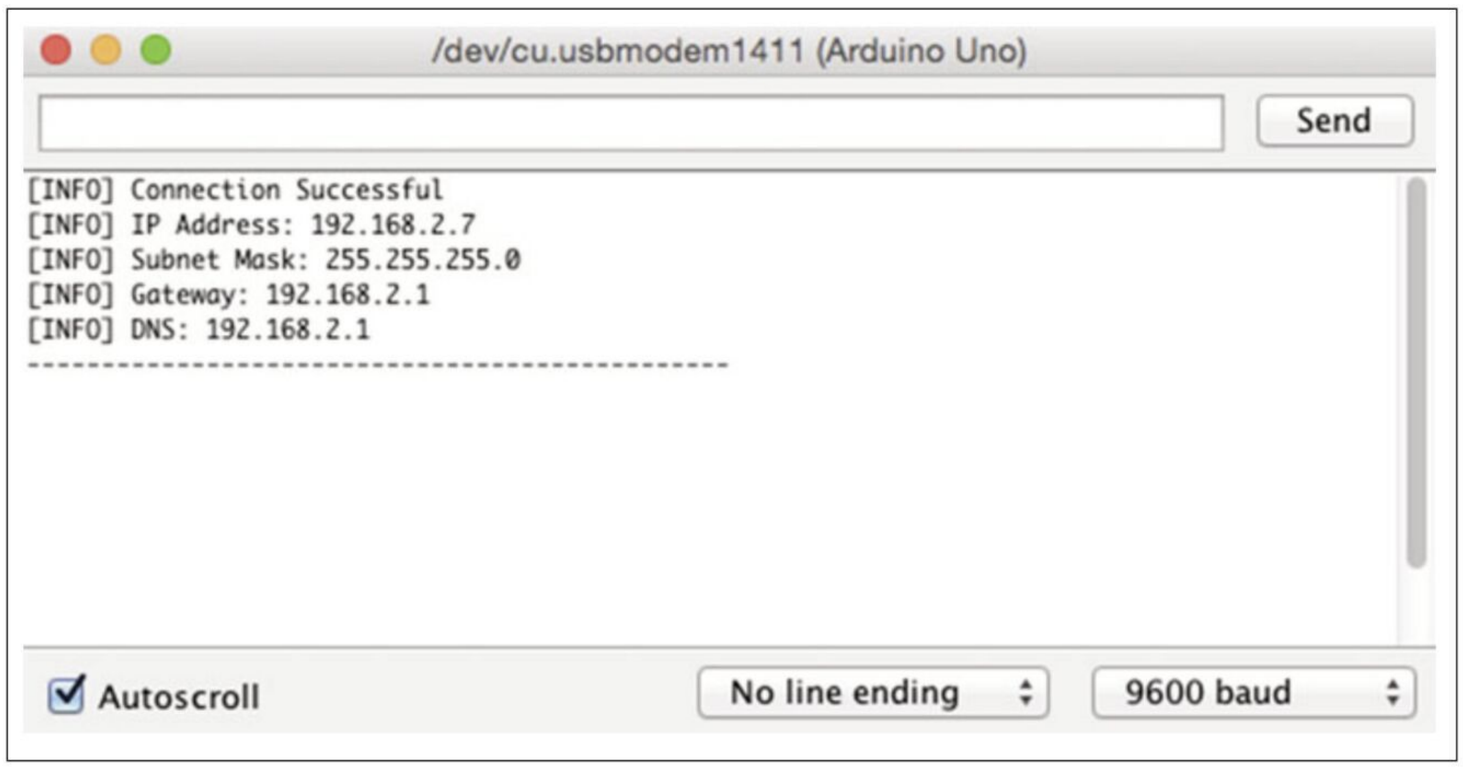


Figure 3: Successfully connected to the internet via Ethernet

2. Connect the Arduino Uno to your computer via the USB port.

Figure 4 shows the circuit design of the Arduino Uno and the Wireless Shield. If everything is connected properly, your Arduino Uno board should look like the one in Figure 5.



Figure 4: Arduino Uno and Wireless Shield



Figure 5: Wireless Shield attached to Arduino Uno

The code

When connecting to the internet via Ethernet, we had used the external library *<Ethernet.h>*. Similarly, for connecting the internet wirelessly, we shall be using the external library *<WiFi.h>*.

> **Note:** This library should come preinstalled with your IDE. If for some reason you run into errors, please download this library from the official GitHub repository of the Arduino Project.

```
#include <SPI.h>
#include <WiFi.h>
```

Next, we shall define the constants and variables required for connecting wirelessly. For connecting with the Wi-Fi, we require the *ssid* and *password* of the Wi-Fi that we shall be using. We shall also create a *WiFiClient* variable for connecting to the internet.

```
char ssid[] = “Write WiFi SSID here”;
char pass[] = “Write WiFi password
here”;

int keyIndex - 0;
int status = WL_IDLE_STATUS;

WiFiClient client;
```

Now we shall define some custom methods for connecting and sending/receiving data over the internet. First up, let us create a method *connectToInternet()* for connecting with the internet.

```
void connectToInternet()
{
      status = WiFi.status();

      if (status == WL_NO_SHIELD)
      {
        Serial.println(“[ERROR] WiFi
Shield Not Present”);
        while (true);
}

while ( status != WL_CONNECTED)
{
      Serial.print(“[INFO] Attempting
Connection - WPA SSID: ”);
      Serial.println(ssid);

      status = WiFi.begin(ssid, pass);
}

Serial.print(“[INFO] Connection
Successful”);
Serial.print(“”);
printConnectionInformation();
Serial.print
ln(“------------------------”);
Serial.println(“”);
}
```

As you can see in the code above, we have called a custom method *printConnectionInformation()* to display the information of our Wi-Fi connection. So let us go ahead and write that method:

```
void printConnectionInformation()
{
      Serial.print(“[INFO] SSID: ”);
      Serial.println(WiFi.SSID());

      // Print Router’s MAC address
      byte bssid[6];
      WiFi.BSSID(bssid);
      Serial.print(“[INFO] BSSID: ”);

      Serial.print(bssid[5], HEX);
      Serial.print(“:”);

      Serial.print(bssid[4], HEX);
      Serial.print(“:”);

      Serial.print(bssid[3], HEX);
      Serial.print(“:”);

      Serial.print(bssid[2], HEX);
      Serial.print(“:”);

      Serial.print(bssid[1], HEX);
      Serial.print(“:”);

      Serial.println(bssid[0], HEX);

      // Print Signal Strength
      long rssi = WiFi.RSSI();
      Serial.print(“[INFO] Signal
Strength (RSSI) : ”);
      Serial.println(rssi);

      // Print Encryption type
```

```
    byte encryption = WiFi.
encryptionType();
    Serial.print("[INFO] Encryption Type
: ");
    Serial.println(encryption, HEX);
    // Print WiFi Shield's IP address
    IPAddress ip = WiFi.localIP();
    Serial.print("[INFO] IP Address : ");
    Serial.println(ip);

    // Print MAC address
    byte mac[6];
    WiFi.macAddress(mac);

    Serial.print("[INFO] MAC Address: ");

    Serial.print(mac[5], HEX);
    Serial.print(":");

    Serial.print(mac[4], HEX);
    Serial.print(":");

    Serial.print(mac[3], HEX);
    Serial.print(":");

    Serial.print(mac[2], HEX);
    Serial.print(":");

    Serial.print(mac[1], HEX);
    Serial.print(":");

    Serial.println(mac[0], HEX);
}
```

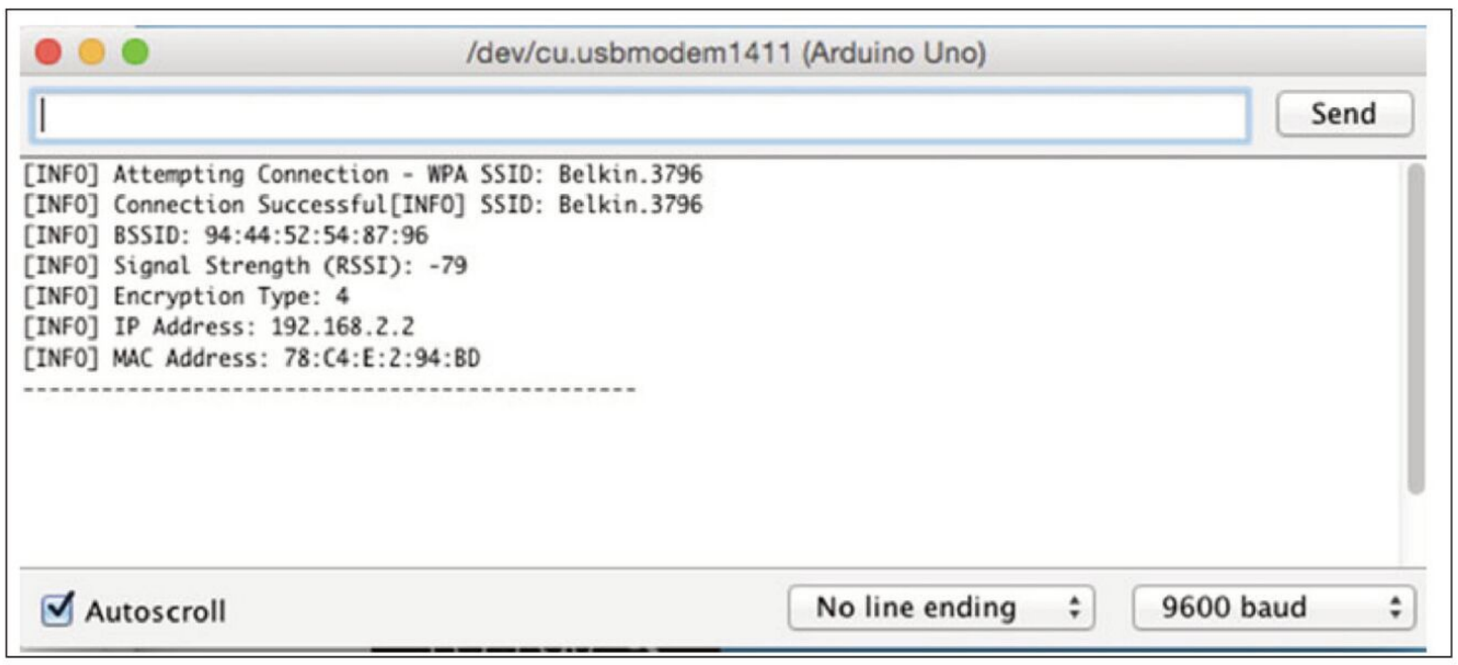


Figure 6: Successfully connected to the internet via Wi-Fi

Finally, let us write down the standard functions, i.e., *setup()* and *loop():*

```
void setup()
{
    Serial.begin(9600);

    // Connect to the internet
    connectToInternet();
}

void loop()
{
    // Nothing to do here
}
```

Hurray! We are done with the Arduino code for connecting to the internet via Wi-Fi. Should there be no errors, the output on your Serial Monitor window should look like the output in Figure 6. END

By: Mir H.S. Quadri

The author is a research analyst with a specialisation in artificial intelligence and machine learning. He is the founder of Arkinfo, which focuses on the research and development of tech products using new age technologies. He shares a deep love for analysis of technological trends and understanding its implications. Being a FOSS enthusiast, he has contributed to several open source projects.

OSFY Magazine Attractions During 2022-23

Month	Theme
March 2022	Security, Network Management and Monitoring
April 2022	Open Source Programming (Languages and tools)
May 2022	DevOps Special
June 2022	AI, Deep learning and Machine Learning
July 2022	Database management and Optimisation
August 2022	Cloud Special: Everything from management to implementation
September 2022	Mobile/Web App Development, Optimisation and Security
October 2022	Blockchain and Open Source
November 2022	Open Source and IoT and Edge
December 2022	Data science, Data security, Data Management,Storage and Backup
January 2023	Docker and containers
February 2023	Open Source on Windows and Best in the world of Open Source (Tools and Services)

A Robot that can Dream, Recognise and Greet Visitors

A first-of-its-kind robot that can recognise a human face and dream—like a human—can now be yours! Interested?

We humans can speak, recognise feelings and emotions, and even dream, which makes us different from machines. But does it make us more intelligent than them? Can machines like robots also acquire this capability?

If yes, how can such intelligent robots be used? As artificial assistants in the workplace, as companions for the elderly and children, for welcoming guests in hotels; these intelligent robots can have multiple uses.

Very few present-day robots are intelligent and they are expensive, costing up to hundreds of thousands of rupees! Keeping this in mind, we are presenting here an intelligent robot that is based on open source tools and libraries. We have used an InMoov robot face mask. You can use the same or any other of your choice. The theme of the robot is shown in Figure 1, while the dreaming robot is shown in Figure 2.

Installation and working

Using Google's open source Deep Dream library, a camera takes a picture and then creates dreams based on it. Artificial intelligence (AI) and Python are used to make the robot recognise your face and talk to you.

Basically, this is the second version of the project that was published in November 2020 issue of *Electronics For You* magazine So, for more information, please see the AI robot details there. This robot is more capable than the previous as it greets the person on recognising and informs us. It also streams the live video over VNC Viewer, as shown in Figure 3. What's more, it has an amazing capability to dream like a human being.

Bill of Materials

Components	Quantity	Description
Raspberry Pi 4	1	2Gb - 4Gb or Higher
Pi Camera With CSI Ribbon	1	5Mp CSI Ribbon Cable
SD Card, 16-32 GB	1	Class 10 or Higher
5V DC Power Adaptor	1	1.5A-2A DC Power Supply
3D Printed Robot Face	1	PLA/ABS 3D Printed Robot



Figure 1: The robot's theme

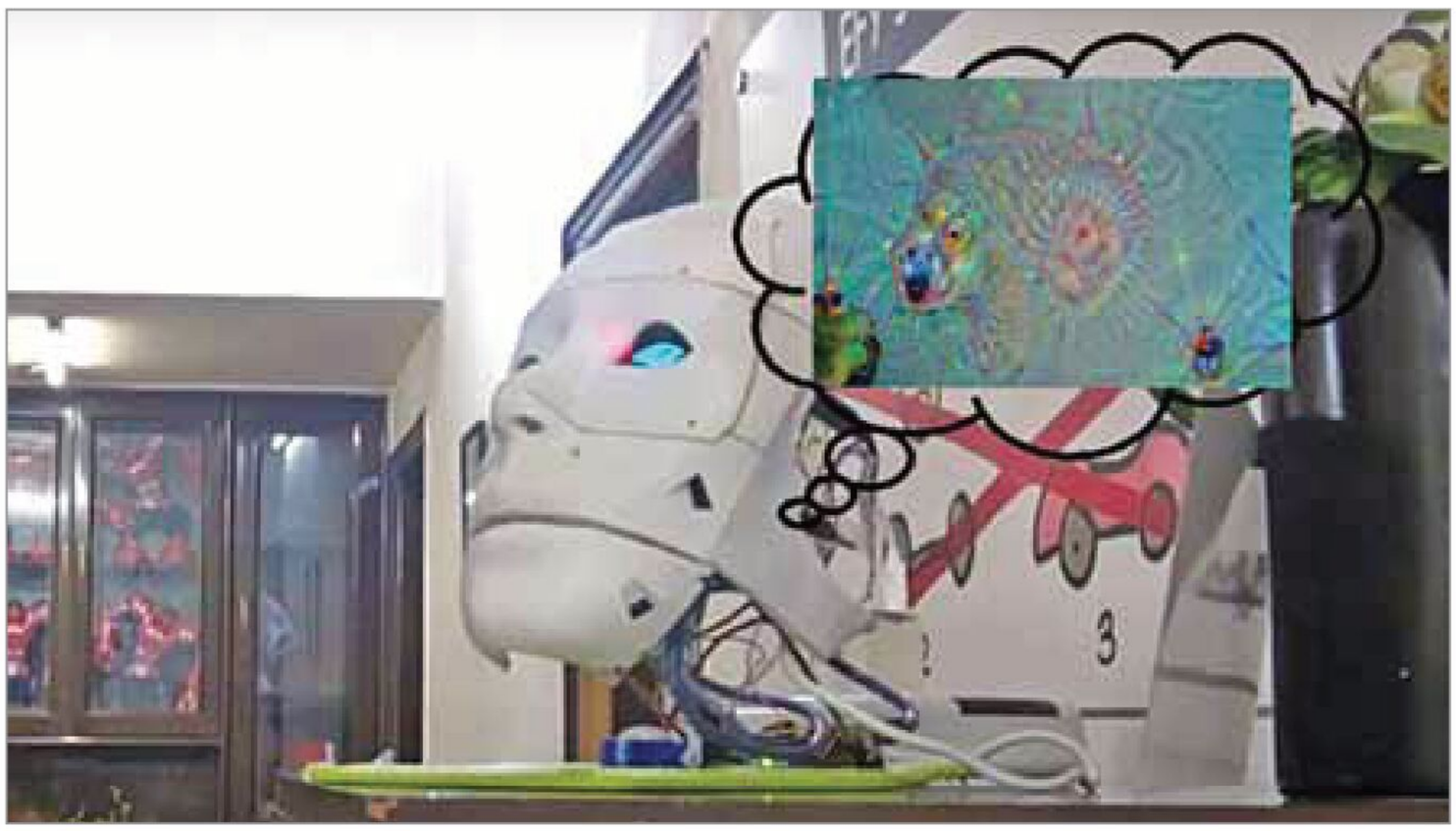


Figure 2: The dreaming robot

To build the project, we require the components mentioned under the Bill of Materials table. We will also need some other parts for making this robot. Check the link *https://www.electronicsforu.com/electronics-projects/design-iot-face-recognition-ai-robot* for that.

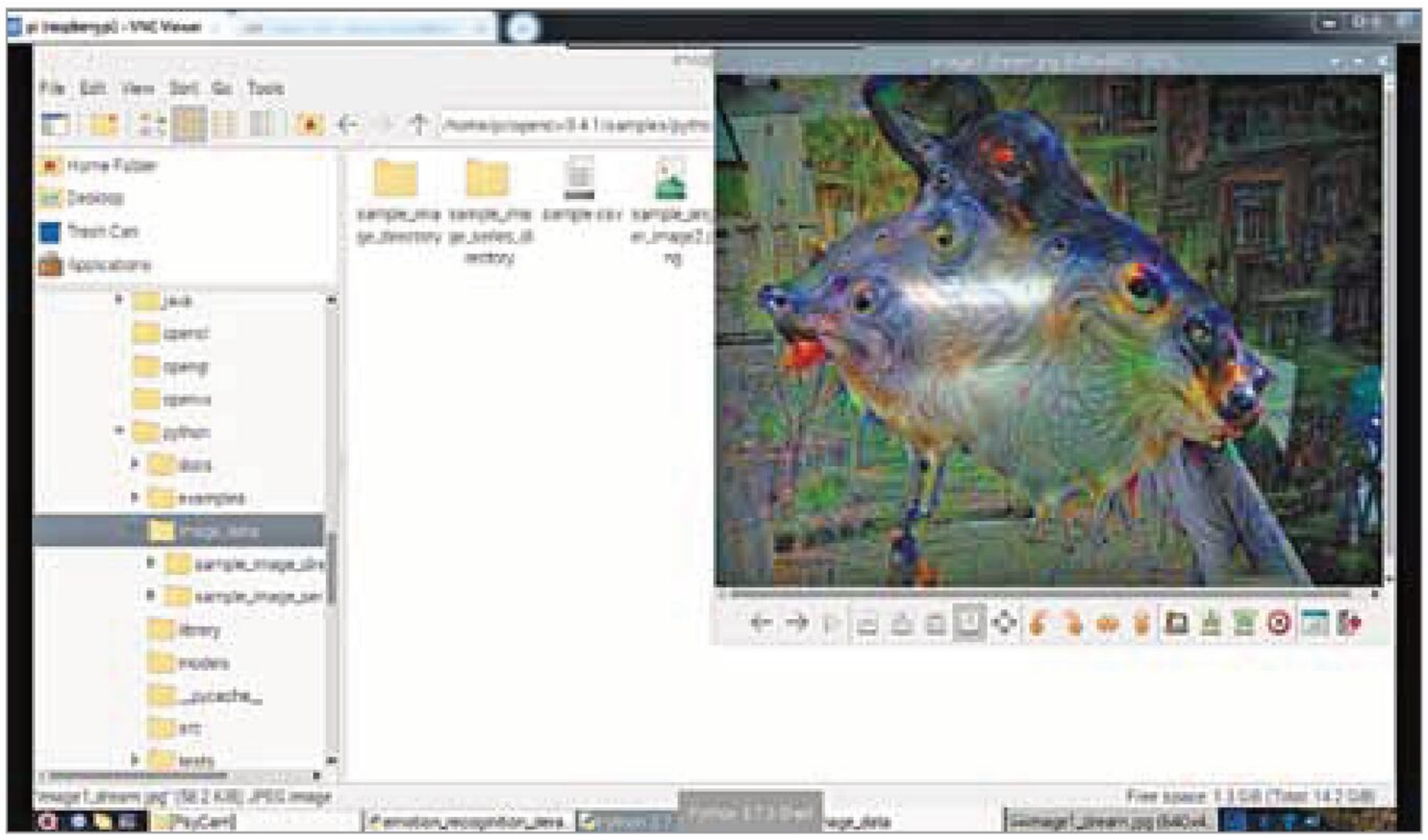


Figure 3: Creating a dream

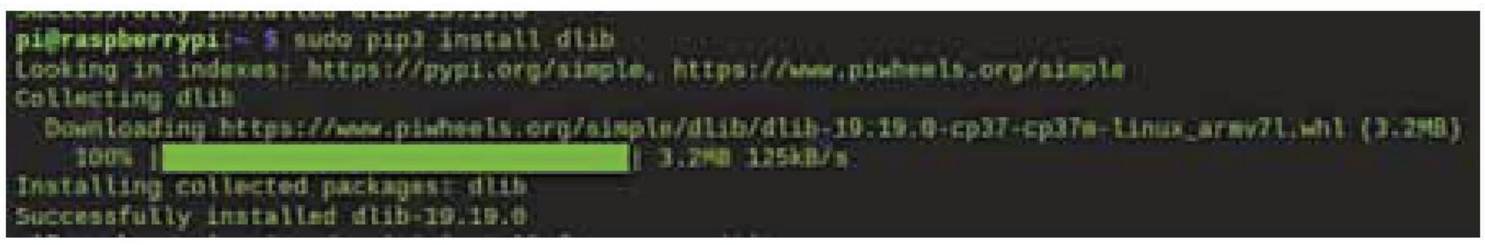


Figure 4: Installing dlib

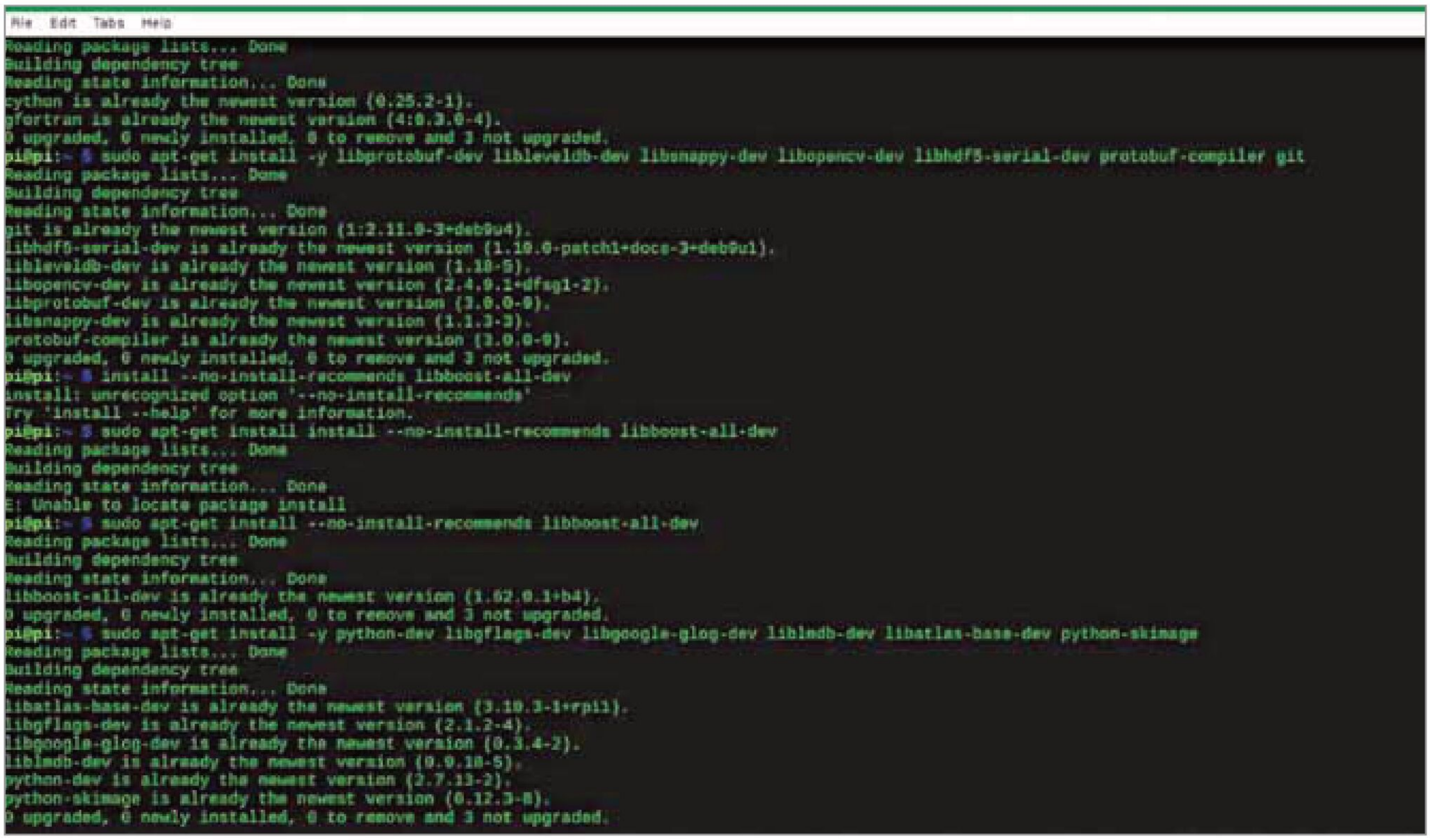


Figure 5: Installing Python modules

This project has two distinct features, for which we need to create a basic setup for both face recognition and deep dreams.

For face recognition requirements/setup, we need to set up the following libraries in Raspberry Pi:

- Keras
- OpenCV
- eSpeak
- Dlib
- Face recognition

To install the above libraries, open the terminal window in Raspberry Pi and run:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install build-essential
   cmake pkg-config
sudo apt-get install libjpeg-dev
 libtiff5-dev libjasper-dev libpng12-
 dev
sudo apt-get install libavcodec-dev
```



Figure 6: Cloning Deedream

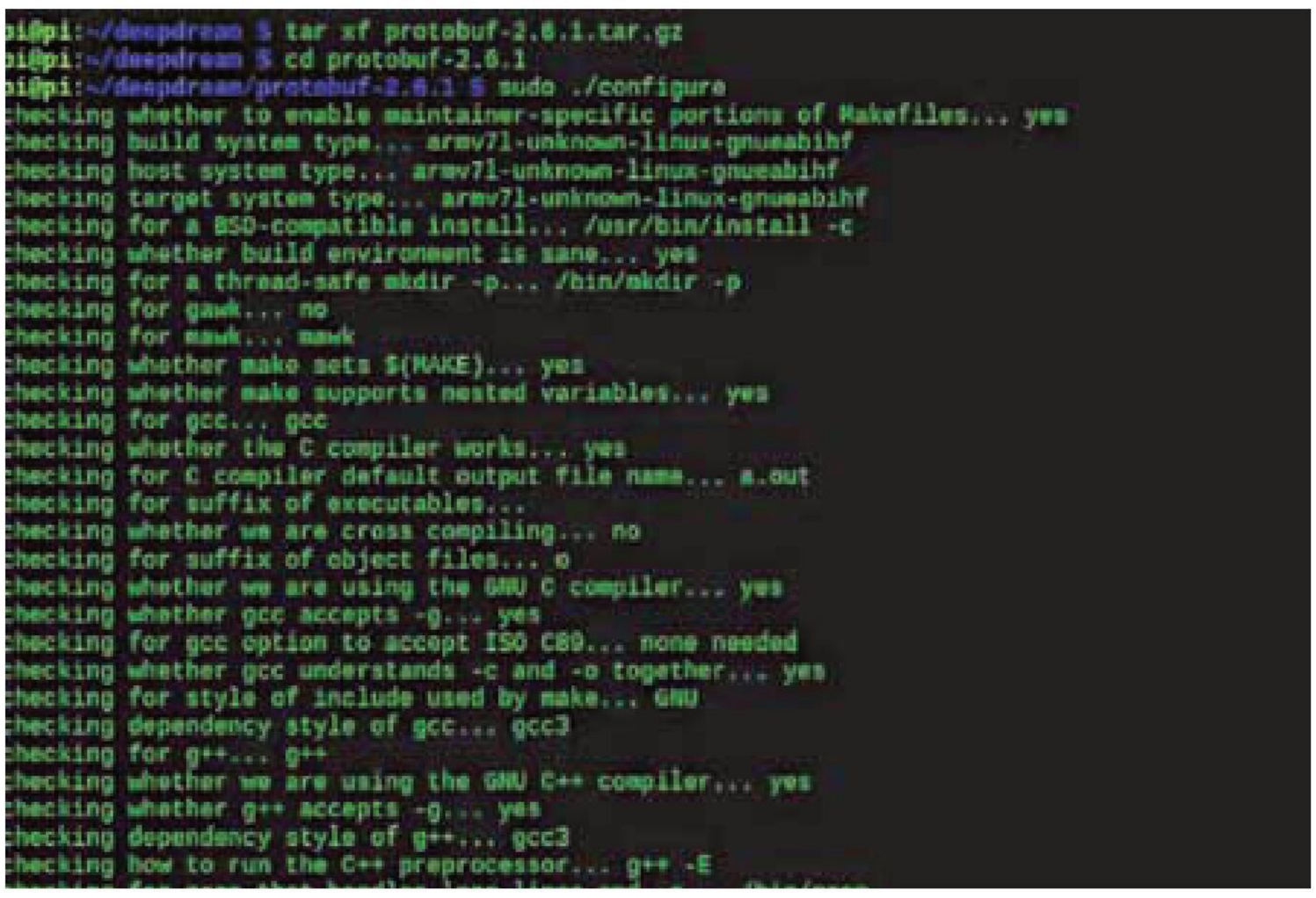


Figure 7: Installing protobuf

```
libavformat-dev libswscale-dev libv4l-
dev
sudo apt-get install libxvidcore-
  dev libx264-dev
sudo apt-get install libgtk2.0-dev
  libgtk-3-dev
sudo apt-get install libatlas-base-
  dev gfortran
```

After that swap the file size. To do so, run the following command:

```
sudo nano /etc/dphys-swapfile
```

Then change the line CONF_SWAPSIZE=100 to CONF_SWAPSIZE=1024

```
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
sudo pip3 install opencvsudo pip3
install
   numpy
wget https://bootstrap.pypa.io/get-pip.
py
pip install dlib
```

Figure 4 shows how to install dlib required for face recognition.

Next, to install the face recognition library, run the following command in the terminal:

```
sudo pip3 install face recognition
```

Run the *sudo apt-get update* command again; with that the basic setup for emotion recognition is done.

For deep dream requirements/setup, install the following modules in Python:

- Caffe
- Protobuf (also known as Protocol Buffers)
- PsyCam

Open your Linux terminal and run the above commands to install the modules (refer Figure 5 through Figure 11). You can also follow the Google Deep Dream official site for installation on Raspberry Pi.

Coding

Having completed the basic setup of the libraries, write the code for real-time face recognition. First, download OpenCV, OpenCV

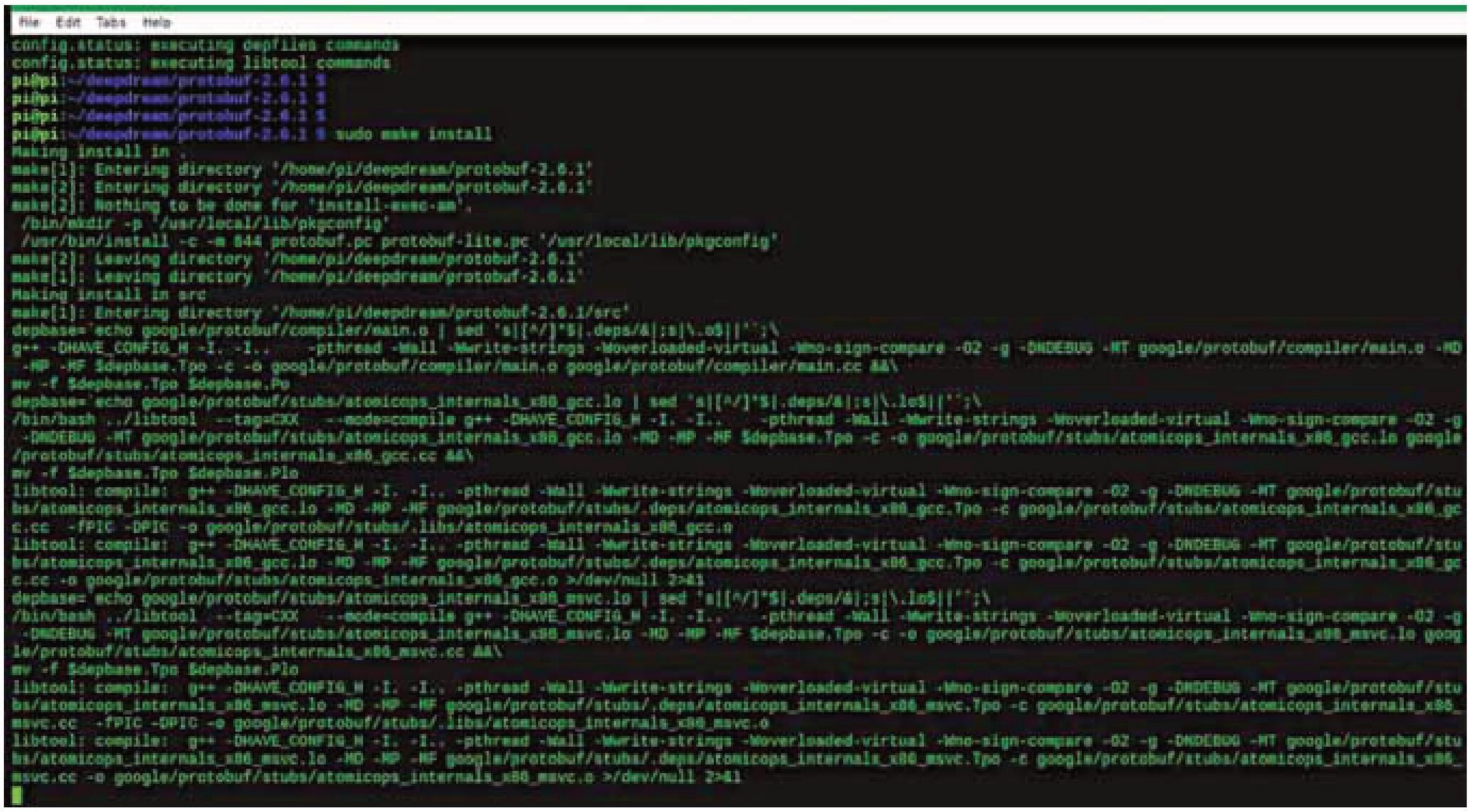


Figure 8: Installing process

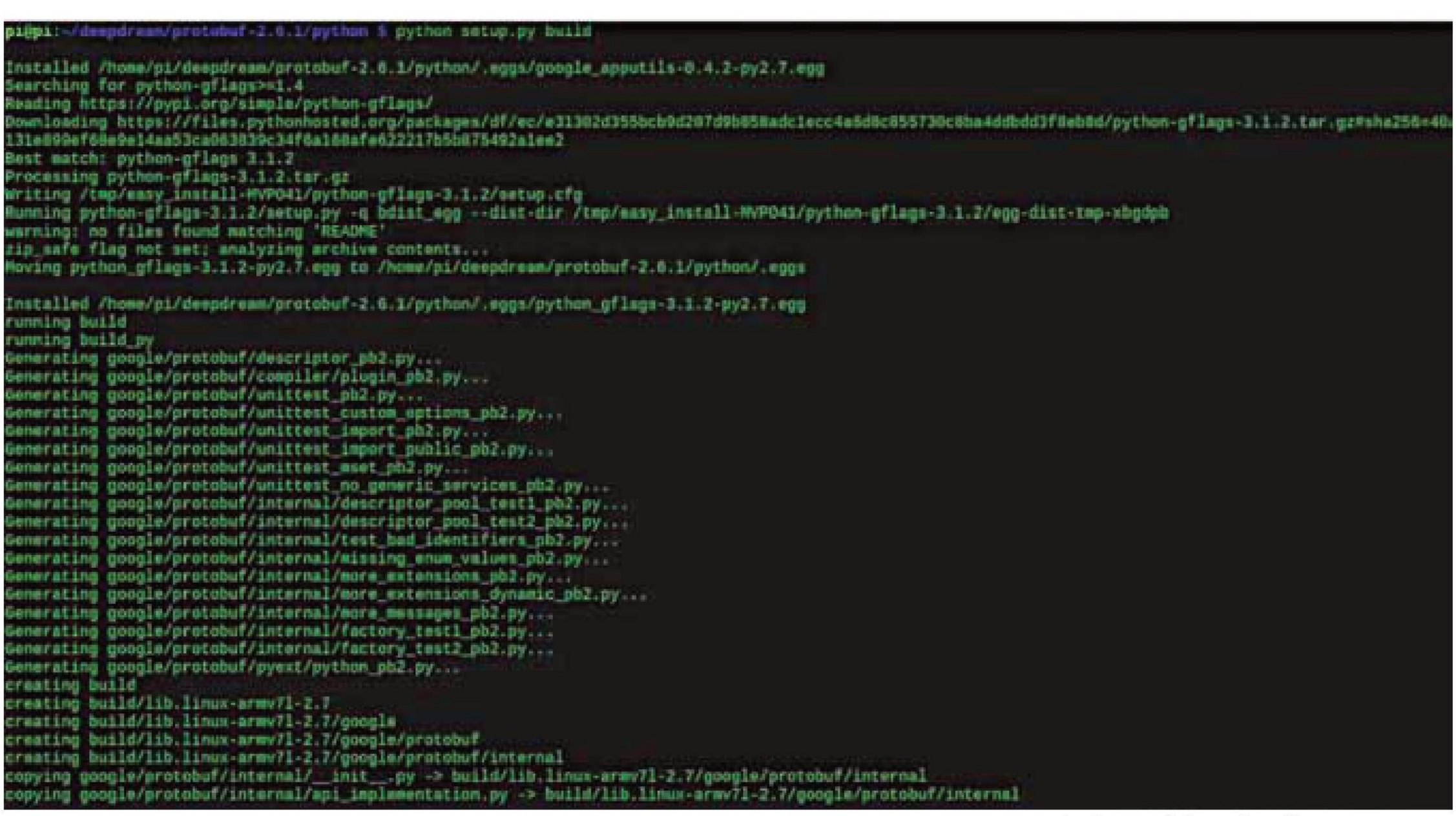


Figure 9: Installation in progress

Contribution, and Face Recognition library folder. To do so, open the terminal window and run the following command in the terminal:

```
git colone https://github.com/opencv/
   opencv
git colone https://github.com/opencv/
   opencv_contrib
```

The above helps recognise the person in front of the robot (known or unknown). In this code, we will import three modules: face recognition, cv2, and numpy. We will create different arrays for recognising faces and names. Make sure to write the image file name of that person for correct face recognition.

In the next part of code, we try to match the face that has been captured by the camera with the array of known faces. If the face matches, the code will run the espeak synthesizer to speak the person's name using the syntax 'espeak.synth ()' as shown in Figure 15.

You may check the Face Recognition Robot project in November 2020 issue of *Electronics For You* to set up the robot head and understand the face recognition system in a better way.

For the deep dream code, download the deep dream PsyCam library using the command:

```
OK
pi@pi:~/deepdream/protobuf-2.6.1/python $
pi@pi:~/deepdream/protobuf-2.6.1/python $ sudo python3 setup.py install
  File "build/bdist.linux-armv7l/egg/google/apputils/run_script_module.py", line 122
    print 'Stub script %s (auto-generated). Options:' % sys.argv[0]
                                                    ^
SyntaxError: invalid syntax

  File "build/bdist.linux-armv7l/egg/google/apputils/file_util.py", line 45
    def Write(filename, contents, overwrite_existing=True, mode=0660, gid=None):
                                                           ^
SyntaxError: invalid token

  File "build/bdist.linux-armv7l/egg/google/apputils/appcommands.py", line 299
    except app.UsageError, error:
                         ^
SyntaxError: invalid syntax

  File "build/bdist.linux-armv7l/egg/google/apputils/basetest.py", line 1063
    0600)
       ^
SyntaxError: invalid token

  File "build/bdist.linux-armv7l/egg/google/apputils/app.py", line 95
    print 'Try --helpfull to get a list of all flags.'
                                                    ^
SyntaxError: Missing parentheses in call to 'print'

zip_safe flag not set; analyzing archive contents...
google.__pycache__.__init__.cpython-35: module references __path__
google.apputils.__pycache__.__init__.cpython-35: module references __path__
google.apputils.__pycache__.setup_command.cpython-35: module references __file__
  File "/home/pi/deepdream/protobuf-2.6.1/python/.eggs/google_apputils-0.4.2-py3.5.egg/google/apputils/app.py", line 95
    print 'Try --helpfull to get a list of all flags.'
                                                    ^
```

Figure 10: *Setup.py* file installing the required modules

```
Testing test_encoding_test
/home/pi/deepdream/protobuf-2.6.1/python/.eggs/google_apputils-0.4.2-py2.7.egg/google/apputils/basetest.py:1617: RuntimeWarning: Trying to access flag test_
mpdir before flags were parsed. This will raise an exception in the future.
  if not os.path.isdir(FLAGS.test_tmpdir):
ERROR:root:Trying to access flag test_tmpdir before flags were parsed.
Traceback (most recent call last):
  File "build/bdist.linux-armv7l/egg/gflags/flagvalues.py", line 535, in __getattr__
    raise exceptions.UnparsedFlagAccessError(error_message)
UnparsedFlagAccessError: Trying to access flag test_tmpdir before flags were parsed.
.
----------------------------------------------------------------------
Ran 2 tests in 0.003s

OK
Testing message_factory_python_test
/home/pi/deepdream/protobuf-2.6.1/python/.eggs/google_apputils-0.4.2-py2.7.egg/google/apputils/basetest.py:1617: RuntimeWarning: Trying to access flag test_
mpdir before flags were parsed. This will raise an exception in the future.
  if not os.path.isdir(FLAGS.test_tmpdir):
ERROR:root:Trying to access flag test_tmpdir before flags were parsed.
Traceback (most recent call last):
  File "build/bdist.linux-armv7l/egg/gflags/flagvalues.py", line 535, in __getattr__
    raise exceptions.UnparsedFlagAccessError(error_message)
UnparsedFlagAccessError: Trying to access flag test_tmpdir before flags were parsed.
...
----------------------------------------------------------------------
Ran 3 tests in 0.225s

OK
Testing message_python_test
/home/pi/deepdream/protobuf-2.6.1/python/.eggs/google_apputils-0.4.2-py2.7.egg/google/apputils/basetest.py:1617: RuntimeWarning: Trying to access flag test_
mpdir before flags were parsed. This will raise an exception in the future.
  if not os.path.isdir(FLAGS.test_tmpdir):
ERROR:root:Trying to access flag test_tmpdir before flags were parsed.
Traceback (most recent call last):
  File "build/bdist.linux-armv7l/egg/gflags/flagvalues.py", line 535, in __getattr__
    raise exceptions.UnparsedFlagAccessError(error_message)
UnparsedFlagAccessError: Trying to access flag test_tmpdir before flags were parsed.
.....................................
----------------------------------------------------------------------
Ran 37 tests in 0.104s

OK
pi@pi:~/deepdream/protobuf-2.6.1/python $
```

Figure 11: Installing Caffe

```
File  Edit  Format  Run  Options  Window  Help
import face_recognition
import cv2
import numpy as np
import time
from espeak import espeak
previous ="unkno"
# This is a demo of running face recognition on live video from your webcam. It'
# other example, but it includes some basic performance tweaks to make things ru
#   1. Process each video frame at 1/4 resolution (though still display it at fu
#   2. Only detect faces in every other frame of video.

# PLEASE NOTE: This example requires OpenCV (the `cv2` library) to be installed
# OpenCV is *not* required to use the face_recognition library. It's only requir
# specific demo. If you have trouble installing it, try any of the other demos t

# Get a reference to webcam #0 (the default one)
video_capture = cv2.VideoCapture(0)


# Load a sample picture and learn how to recognize it.
ash_image = face_recognition.load_image_file("ash.jpg")
ash_face_encoding = face_recognition.face_encodings(ash_image)[0]

# Load a second sample picture and learn how to recognize it.
alhil_image = face_recognition.load_image_file("Alhil.jpg")
alhil_face_encoding = face_recognition.face_encodings(alhil_image)[0]

# Load a sample picture and learn how to recognize it.
atul_image = face_recognition.load_image_file("Atul.jpg")
atul_face_encoding = face_recognition.face_encodings(atul_image)[0]

# Load a second sample picture and learn how to recognize it.
gaurav_image = face_recognition.load_image_file("Gaurav.jpg")
gaurav_face_encoding = face_recognition.face_encodings(gaurav_image)[0]
```

Figure 12: Face recognition code

```
face_encodings = []
face_names = []
process_this_frame = True

while True:
    # Grab a single frame of video
    ret, frame = video_capture.read()

    # Resize frame of video to 1/4 size for faster face recognition processing
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

    # Convert the image from BGR color (which OpenCV uses) to RGB color (which f
    rgb_small_frame = small_frame[:, :, ::-1]

    # Only process every other frame of video to save time
    if process_this_frame:
        # Find all the faces and face encodings in the current frame of video
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_l

        face_names = []
        for face_encoding in face_encodings:
            # See if the face is a match for the known face(s)
            matches = face_recognition.compare_faces(known_face_encodings, face_
            name = "Unknown"


            # # If a match was found in known_face_encodings, just use the first
            # if True in matches:
            #     first_match_index = matches.index(True)
```

Figure 14: Open Cv processing image

```
File  Edit  Format  Run  Options  Window  Help
    rahul_face_encoding,
    shipra_face_encoding,
    vinay_face_encoding,
    vinita_face_encoding,
    parul_face_encoding,
    tanyaaneja_face_encoding,
    deba_face_encoding,
    tanya_face_encoding,
    reshu_face_encoding,
    astha_face_encoding,
    akhilesh_face_encoding,


]
known_face_names = [
    "Ashwini kumar",
    "Akhil ",
    "Atul",
    "gaurav lo",
    "mukul",
    "shavita",
    "rahul",
    "shipra",
    "vinay",
    "vinita",
    "parul",
    "tanya ",
    "deba",
    "tanya",
    "reshu",
    "astha",
    "akhilesh"
]
```

Figure 13: Database of known face names

```
File  Edit  Format  Run  Options  Window  Help
            #     name = known_face_names[first_match_index]

            # Or instead, use the known face with the smallest distance to the n
            face_distances = face_recognition.face_distance(known_face_encodings
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = known_face_names[best_match_index]
                if previous != name:
                    previous=name
                    print(name)
                    #espeak.synth("Hey I Can Identify your face")
                    #espeak.synth("Your name is")
                    #espeak.synth(name)
                    espeak.set_voice("En")
                    espeak.set_voice("whisper")
                    espeak.synth("Hey hello ")
                    espeak.synth(name)
                    espeak.synth("How are you ")
                    espeak.synth("have a nice day")

                else:
                    print("old same")


            face_names.append(name)


    process_this_frame = not process_this_frame


    # Display the results
    for (top, right, bottom, left), name in zip(face_locations, face_names):
        # Scale back up face locations since the frame we detected in was scaled
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4
```

Figure 15: Greeting code part to call and say when person is detected

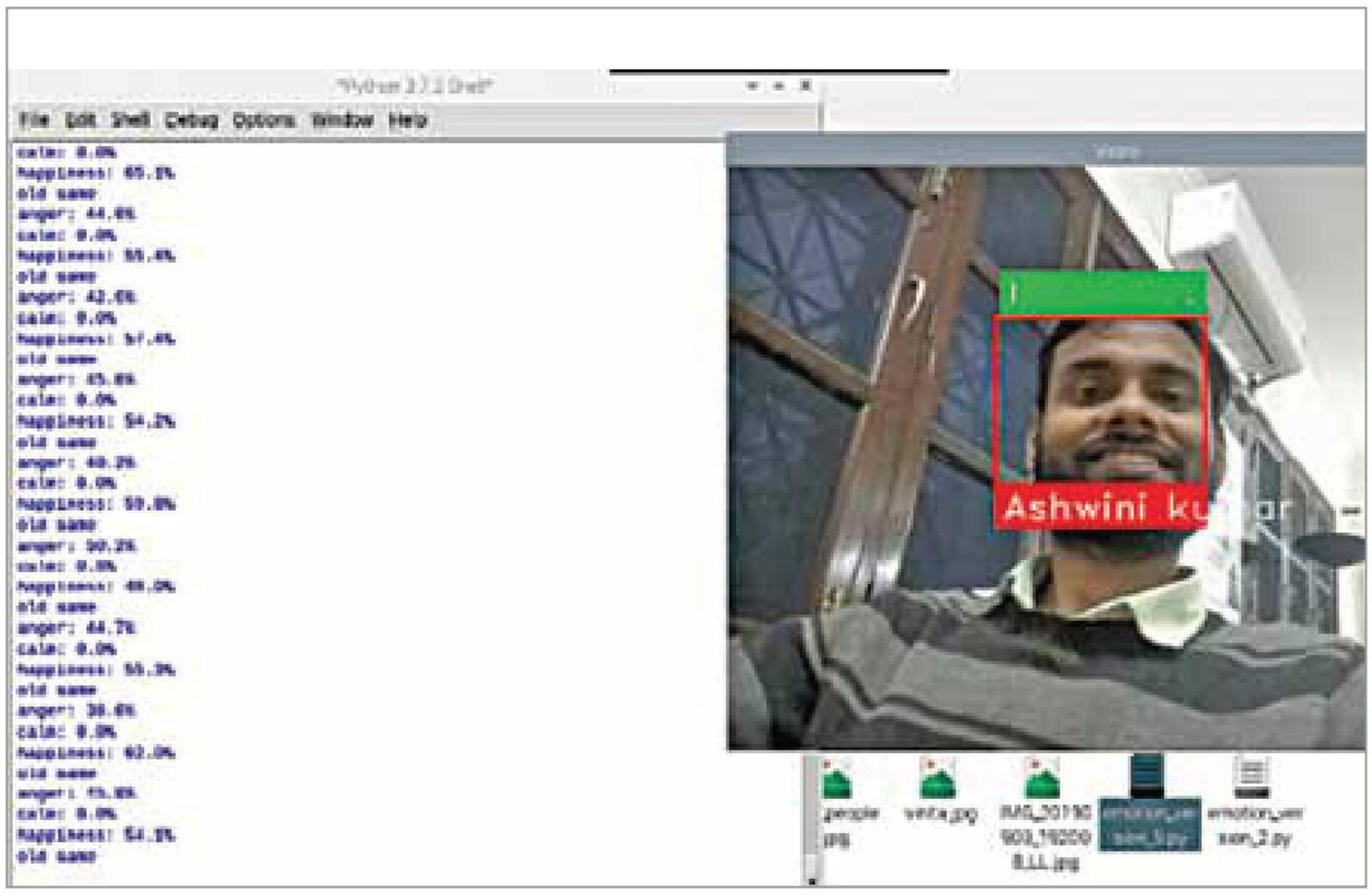


Figure 16: Face detection



Figure 17: Random robot dream



Figure 18: Robot's dream generated from the saved incident captured by camera



Figure 19: Original picture



Figure 20: The robot's dream generated from the picture in Figure 19

```
git clone https://github.com/JoBergs/
   PsyCam
```

Then open the PsyCam folder and change the file path for dream creating image in *utlities.py* to the same path where the captured frame of *facrecog.py* image was saved. In the same folder, create a new Python file and then a script that runs the *testpsycam.py* file in certain intervals of time and makes it go to sleep mode, that is, off/on.

To see the dream image created by the robot, open the dream folder that is situated in the PsyCam folder. This completes the coding.

Connection and testing

Connect the Raspberry Pi Type-C USB port with 5V cable. Next, connect the Raspberry Pi camera using a ribbon cable to the Raspberry Pi camera port. Connect the eyes OLEDs and other parts as described in the earlier AI robot article published in November 2020 issue.

After setting and saving all the codes, test various features of the robot. Run the *face.py* and *run.py* scripts, and wait for a few seconds. The screen will then start showing the face and the robot will start talking according to the face it detects (see Figure 16).

To check the dream of the robot, run the code *pycam.py*. The robot will start capturing the pics and start creating a dream. You may open the dreams folder and see the saved picture of the robot's dream (see Figure 17 through Figure 20).

Note: The source code of this project is available for free download at *source.efymag.com*

END

By: Ashwini Kumar Sinha

The author is a technology enthusiast at *EFY*

The article was originally published in the September 2022 issue of Electronics For You.

EFYGROUP
YOURS SINCE 1969

Thank You IIT Madras

It was in your hostel room that the idea was conceived



The first issue in Jan '69



The Jan '22 Issue

Celebrating 54 years in TECH media

Magazines
- Electronics For You
- Electronics Bazaar
- Open Source For You

Portals
- www.electronicsforu.com
- www.opensourceforu.com
- www.eleb2b.com
- www.electronicsb2b.com

Events
- India Electronics Week
  - IOTshow.in
  - EFY Expo
  - Test & Measurement India
- Open Source India
- Techworldcongress.com

EFY Enterprises Pvt Ltd
D-87/1, Okhla Industrial Area, Phase-1, New Delhi-110020

• Phone: +91-98111-55335
• Email: info@efy.in • Website: www.efy.in

R N I No. DELENG/2012/49440, Mailed on 27/28th of Advance month
Published on 27th November 2022
Delhi Postal Regd. No. DL(S)-01/3443/2022-24
Licenced to Post without Pre-Payment Licence No U(S)-56/2022-24
Posted at LPC Delhi RMS Delhi 110006

India's No.1 Electronics Magazine
RANKING AS PER ELCINA-IMRB REPORT

"I started reading it when I was a student... ...and I am still reading it, as a student"

—CEO, Design House



electronics FOR YOU
YOURS SINCE 1969

To Subscribe:
https://subscribe.efy.in

OR

Scan This Code

For any query, call: +91-98111-55335 or email: support@efy.in

