

OpenSource

Volume: 05 | Issue: 03 | Pages: 108 | December 2016

THE COMPLETE MAGAZINE ON OPEN SOURCE

For You

Volume: 05 | Issue: 03 | Pages: 108 | December 2016

ForYou

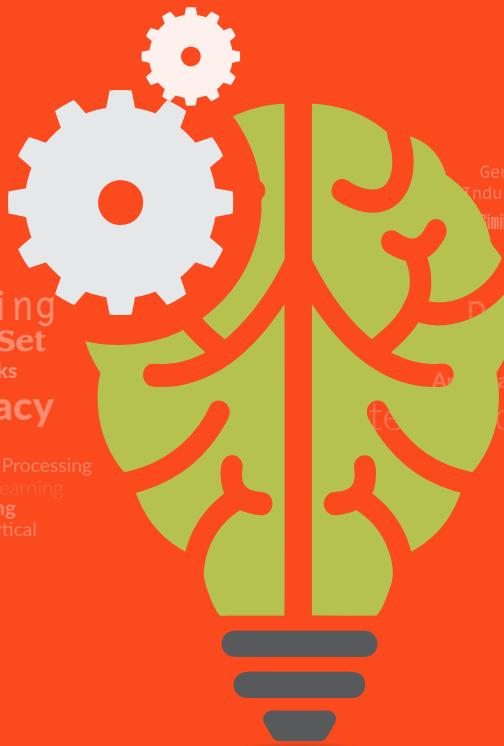
An **EFY GROUP** Publication

Improving The IQ Of **COMPUTING SYSTEMS**

Understanding The Basics Of Machine Learning

An Introduction To H2O

Data Mining Data Set Artificial Neural Networks Intelligence Accuracy



“People Are Now Even Doing Machine Learning In JavaScript”

–An Interview With **Brendan Eich**, The Creator Of JavaScript

Machine Learning: The Julia Approach

An Interview With **Pradeep Chandru,** **Founder And CEO,** **Mafree**





MARCH 2-4 '17. BIEC. B'LORE.

Supported By
Ministry of Electronics &
Information Technology,
Government of India

PROFIT FROM



AN EVENT FOR THE **CREATORS**,
THE **ENABLERS** AND THE **CUSTOMERS** OF IOT.

IoTshow.in is India's biggest expo-cum-conference on Internet of Things.

Its first edition in 2016 was voted as world's top IoT event at Postscapes.com!

Let's make the 2017 edition even **BIGGER**—and put India at the centre of the IoT global map.

Visitor Registration: www.iotshow.in. Special Offers for first 1000 Registrants!

CO-LOCATED WITH:

INDIA
ELECTRONICS
WEEK

BROUGHT TO YOU BY:



MORE INFO:

Web: www.iotshow.in
Email: support@efy.in
Bulk Registration: iew@efy.in
Tel: +91-11-40596605

Be Up to Date with Technology Strengthen Reading Habits

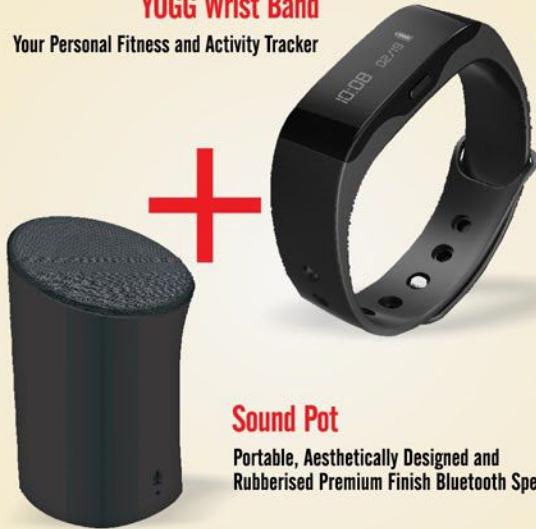
And, enjoy these FREE Gifts too...

You & Me

YOGG Wrist Band

Your Personal Fitness and Activity Tracker

Worth
₹2999



Worth
₹1599

Sound Pot

Portable, Aesthetically Designed and
Rubberised Premium Finish Bluetooth Speaker



Wireless
Headphone
Worth ₹1999



USB Car
Charger
Worth ₹499

5200mAh
Power Bank
Worth ₹1999



ORDER FORM

Magazine	Duration	Cover Price (₹)	You Pay (₹)	Gifts For You & Me	Please Tick Any One	Gift for Techies
Open Source For You with DVD	1 Year	1440	1440	<input type="checkbox"/> 5200mAh Power Bank - worth Rs 1999	<input type="checkbox"/>	Arduino Uno worth Rs 890 + Smart WiFi Module - worth Rs 725
	2 Years	2880	2880	<input type="checkbox"/> Wireless Headphone - worth Rs 1999 + USB Car Charger - worth Rs 499	<input type="checkbox"/>	IoT Starter Kit - worth Rs 2000
	5 Years	7200	7200	<input type="checkbox"/> YOGG WristBand - worth Rs 2999 + Sound Pot - worth Rs 1599	<input type="checkbox"/>	Raspberry Pi 3 - worth Rs 3300

Free e-zine Access With Every Subscription | To Subscribe Online, visit: <http://subscribe.efyindia.com>



Name _____ Designation _____ Organisation _____

Mailing Address _____

City _____ Pin Code _____ State _____ Phone/Mobile _____ Email _____

Subscription No. (for existing subscribers only) _____ I would like to subscribe Open Source For You 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)

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: 011-26810601-03 | Fax: 011-26817563 | e-mail: info@efy.in | www.efy.in

EFY GROUP
Technology Drives Us

Terms & conditions: # This offer is applicable for new subscriptions and renewal by existing subscribers. # The rates are valid for subscribers within India only. # Please allow 4-6 weeks for processing your subscription and 6-8 weeks to send your gifts # Replacement copies will be sent only if intimation of damaged / non-receipt of copies is received within 15 days of publication # If you are not satisfied with the magazine or our services, we shall refund the balance amount after three months. # Disputes, if any, are subject to exclusive jurisdiction of competent courts and forums in Delhi/New Delhi only.

Contents

Admin

- 31 Apache Spark: The Ultimate Panacea for the Big Data Era
- 39 Varnish: A Performance Booster for Web Applications

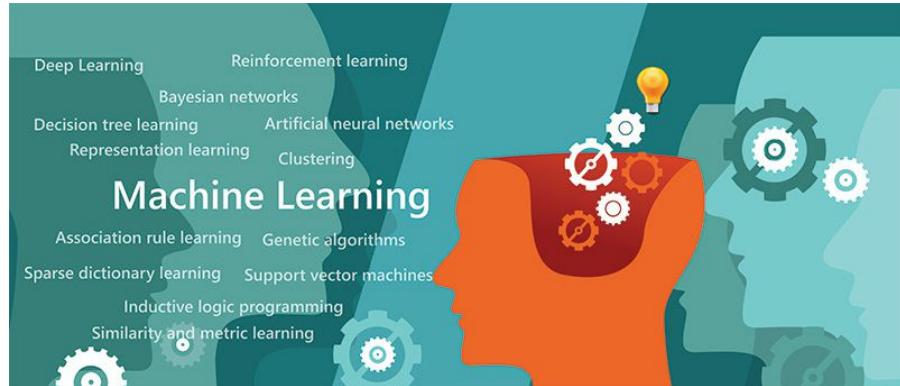
Developers

- 41 REST API Development Using Django Tastypie Framework
- 46 Create Your Own Java Based Chat Robot
- 49 Creating a Barcode Generator in App Inventor 2
- 52 An Introduction to Weka
- 55 Analysing Sentiments with NLTK
- 57 An Introduction to Google Machine Learning APIs
- 61 Machine Learning: The Julia Approach
- 64 Ruby on Rails: A Powerful Open Source Web Framework for Beginners
- 68 The Best Open Source Machine Learning Frameworks
- 72 Autolt: An Open Source Software Testing Tool for Windows
- 77 Does Your Mobile App Work Without an Internet Connection?
- 82 An Introduction to H2O and its Relation with Deep Learning
- 89 regmap: Reducing the Redundancy in Linux Code



REGULAR FEATURES

- | | | | |
|----|--------------|-----|--------------------|
| 06 | New Products | 101 | Editorial Calendar |
| 08 | FOSSBytes | 102 | Tips & Tricks |



Machine Learning Basics for Newbies 98



Brendan Eich,
the creator
of JavaScript

"People are now even
doing machine learning
in JavaScript" | 26

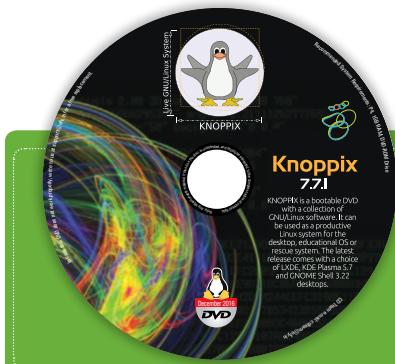


Pradeep Chandru,
founder and CEO
of Mafiree

"We come with
ample expertise
in MySQL" | 104

FOR U & ME

93 A Few Tips on Vi/Vim
Editor for Linux Newbies



Columns

15 Exploring Software:
Programming with Objects
in Alice 3

29 CodeSport

DVD OF THE MONTH

Experience Linux in live mode.

- KNOPPIX 7.7.1

| 106

EDITOR
RAHUL CHOPRA

EDITORIAL, SUBSCRIPTIONS & ADVERTISING
DELHI (HQ)
D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020
Ph: (011) 26810602, 26810603; Fax: 26817563
E-mail: info@efy.in

MISSING ISSUES
E-mail: support@efy.in

BACK ISSUES
Kits 'n' Spares
New Delhi 110020
Ph: (011) 26371661, 26371662
E-mail: info@kitsnspares.com

NEWSSTAND DISTRIBUTION
Ph: 011-40596600
E-mail: efycirc@efy.in

ADVERTISEMENTS
MUMBAI
Ph: (022) 24950047, 24928520
E-mail: efymum@efy.in

BENGALURU
Ph: (080) 25260394, 25260023
E-mail: efybblr@efy.in

PUNE
Ph: 08800295610/09870682995
E-mail: efypune@efy.in

GUJARAT
Ph: (079) 61344948
E-mail: efyahd@efy.in

CHINA
Power Pioneer Group Inc.
Ph: (86 755) 83729797, (86) 13923802595
E-mail: powerpioneer@efy.in

JAPAN
Tandem Inc., Ph: 81-3-3541-4166
E-mail: tandem@efy.in

SINGAPORE
Publicitas Singapore Pte Ltd
Ph: +65-6836 2272
E-mail: publicitas@efy.in

TAIWAN
JK Media, Ph: 886-2-87726780 ext. 10
E-mail: jkmedia@efy.in

UNITED STATES
E & Tech Media
Ph: +1 860 536 6677
E-mail: veronique.lamarque@gmail.com

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 © 2016. 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 favor of **EFY Enterprises Pvt Ltd**.
Non-receipt of copies may be reported to support@efy.in—do mention your subscription number.



NEW PRODUCTS

Price:
₹ 4,990 for JBL Clip 2 and
₹ 29,990 for Go+Play

Two portable Bluetooth speakers from Harman

Harman International Industries, a manufacturer of home and car audio equipment, has launched two Bluetooth speakers in India, namely, JBL Clip 2 and Go+Play Mini.

The waterproof JBL Clip 2 is a next-generation Bluetooth speaker that comes with a high-quality, powerful sound system and increased playback time. The rugged device is designed with a durable fabric casing along with a smart carabiner that can be attached with almost everything.

This easily portable device can be used for any outdoor or indoor activity.

The Harman Kardon Go+Play Mini is the smaller version of the previously launched Go+Play.

It features wireless Bluetooth streaming with dual sound and a microphone conferencing system for natural sound even in noisy environments. The device comes with a versatile stainless steel handle. It has a built-in rechargeable battery offering up to eight hours of non-stop music. The device also works as a power bank, as users can charge their smartphones and other devices via the USB charge port.

The Harman Kardon JBL Clip 2 and Go+Play Mini are available online and at retail stores.

Address: Harman India, Prestige Technology Park, 4th Floor – Jupiter (2A) Block, Marathahalli Ring Road, Bengaluru 560103; **Ph:** +918043306300; **Fax:** +91 80 40976806

BlackBerry's secure Android smartphone

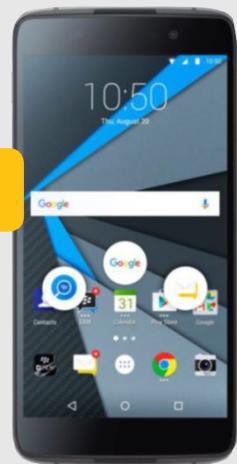
Multinational mobile hardware company, BlackBerry, has launched its latest smartphone in India, the DTEK50. The device sports a 13.2cm (5.2 inch) full HD (1080 x 1920 pixel) display with a pixel density of 424ppi.

The single-SIM (micro-SIM) device is powered by a 1.2GHz octa-core Qualcomm Snapdragon 617 SoC. It runs on Android 6.0.1 Marshmallow and is coupled with 3GB of RAM. The company claims the device is the world's most secure smartphone.

It comes with 16GB inbuilt storage expandable up to 2TB via a microSD card, and is powered by a 2610mAh non-removable battery. The camera features of the smartphone include a 13 megapixel rear camera with PDAF, dual-tone LED flash with a 6-element lens and an f/2.0 aperture.

It also has an 8 megapixel fixed focus camera with an 84 degree wide

Price:
₹ 21,990



angle lens and an f/2.2 aperture.

The DTEK50 offers connectivity options like 4G LTE, Wi-Fi 802.11ac, FM radio, Bluetooth v4.2, GPS/A-GPS and NFC. It also offers additional features like an accelerometer, ambient light sensor, gyroscope, magnetometer and a proximity sensor.

The BlackBerry DTEK50 is available online and at retail stores.

Address: BlackBerry India, No-76, Udyog Vihar 1, Gurugram, Haryana; **Ph:** +91-124-4648800

Rugged hard disk from ADATA

Price:
₹ 4,550 for 1TB and
₹ 7,880 for 2TB



Taiwanese memory and storage manufacturer, ADATA Technology Co., has launched its HD700 hard drive, which has a storage capacity of 1TB and 2TB. The device is rugged and easy-to-carry. Its silicone sheath cover protects it from dust and moisture.

The device is designed with cut-outs on the silicone cover to connect it with a computer.

The thick rubber flap covers the microUSB 3.0 port. The HD700 is

thicker than most external hard drives, but is easily portable. It offers a transfer speed of 5Gbps, which is the theoretical limit of USB 3.0.

The ADATA HD700 is available in blue and black, online and at retail stores.

Address: ADATA Technology Co., 215 Atrium, Office No. 219, C Wing, 2nd Floor, Andheri Kurla Road, Andheri (East), Mumbai 400059; **Fax:** (022) 6758-7045

Solar chargeable power bank from UIMI



Price:
₹ 1,900

Electronic devices manufacturer, UIMI, has launched its 6000mAh UIMI U3 power bank.

This is the company's first power bank to be made in India; it supports

solar charging as well as regular charging via AC power sockets and has a rubber finish.

The power bank comes with a single input port and dual output USB port for charging two devices simultaneously. It sports a 2.4W LED panel light.

The water- and dust-proof device is available in deep sky blue and lime green colour options via online stores.

Address: UIMI Technologies, F-16, Sector-6, Noida, Uttar Pradesh 201301; **Ph:** 91-120-4552102

Price:
₹ 5,490



Sony's extra-bass wireless headset

Japanese tech giant, Sony, has launched another affordable wireless headset in India, the MDR-XB50BS. Perfect for sports enthusiasts, the device connects via Bluetooth and NFC with up to 8.5 hours of music playback time.

The IPX4 splash-proof device enables users to keep the music on even during a light drizzle. It offers hands-free calling with HD voice support via a built-in microphone. The right side ear piece of the Sony MDR-XB50BS is designed with volume rocker keys, which also act as the track change button.

There is another multi-function button that offers the power on/off, play/pause and call receive/reject features. The Sony MDR-XB50BS is available in black, blue and red via online stores.



Price:
₹ 17,499

video calling. The device is preloaded with a multi-language keyboard, which supports 21 regional languages.

The Slide Brace XI comes with a huge 7800mAh battery with up to seven hours and 30 minutes video playback on one charge. It has a thick cylindrical stand at the bottom, which allows the device to rotate and stand upright on any surface.

The iBall Slide Brace X1 4G tablet is available in 'bronze gold' online and at retail stores.

Address: iBall, U-202, Third Floor, Pillar No. 33, Near Radhu Palace, Laxmi Nagar Metro, New Delhi – 110092; **Ph:** 011-26388180

Multi-feature tablet from iBall

Consumer electronics company, iBall, has launched its latest tablet, the iBall Slide Brace X1, with a number of new features. The company claims this tablet is stronger and faster than its predecessors.

It features a 25.5cm (10.1 inch) capacitive multi-touch IPS display with a resolution of 1280x800 pixels.

Powered by a 1.3GHz octa-core ARM Cortex-A53 processor with a MaliT720 GPU, the device offers 2GB of RAM. It runs on Android 6.0 Marshmallow out-of-the-box and comes with inbuilt storage of 16GB, expandable up to 64GB using a microSD card. The device features connectivity options like 4G, VoLTE, Wi-Fi, Bluetooth, micro-USB, GPS/A-GPS, and OTG support.

The tablet is equipped with an 8 megapixel rear camera, auto-focus with LED flash, along with a front 5 megapixel camera for selfies and

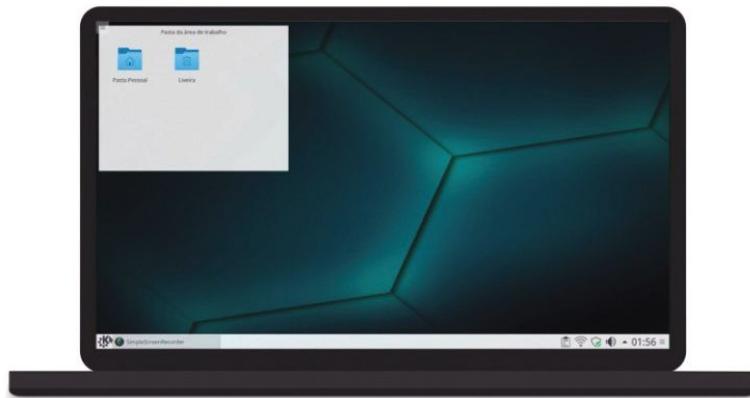
Address: Sony India, No-A-31, Mohan Co-operative Industrial Estate, Mathura Road, New Delhi – 110044; **Ph:** 011-66006600

The prices, features and specifications are based on information provided to us, or as available on various websites and portals. OSFY cannot vouch for their accuracy.

Compiled by: Aashima Sharma

OpenSUSE Tumbleweed gets latest Flatpak framework

SUSE has announced a new update for OpenSUSE Tumbleweed that comes with the most recent Flatpak framework. The newest version of Tumbleweed also includes some other updated packages to deliver an enhanced experience.



To celebrate Halloween with developers, the OpenSUSE Tumbleweed 20161028 snapshot comes with Flatpak 0.6.13 to offer desktop applications on the Linux environment. There is also OSTree 2016.12, which offers a layer for deploying bootable file system trees and managing bootloader configurations.

The SUSE team has additionally included packages such as Mozilla Firefox 49.0.2 and Frameworks 5.27.0, along with new MIME type icons. The platform also brings updates to openSUSE-specific packages, including YaST2-storage 3.1.105 and YaST2-http-server 3.2.1.

Apart from this, SUSE has released Tumbleweed snapshot 20161101 with Hexchat 2.12.3, Wine 1.9.22 and Nmap 7.31. The four other versions after the snapshot 20161028 have also received new treatments like some sub-packages for AppArmors and dbus-1-glib, and Kiwi OS image builder version 7.04.8.

Developers can install the newest OpenSUSE Tumbleweed snapshot on their systems immediately. It can be downloaded through the OpenSUSE factory.

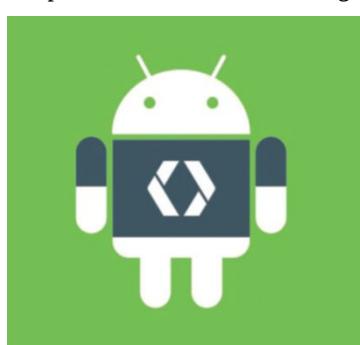
Google ends support, development for Eclipse Android Developer Tools (ADT)

Google has finally departed from the way in which it developed apps for its open source platform by ending its support and stopping development work for the Eclipse ADT. Instead, the search giant is now focusing on Android Studio, which

debuted as the official IDE for the Android ecosystem back in May 2013.

The new development happened following the arrival of Android Studio 2.2.2 last month.

“There has never been a better time to switch to Android Studio and experience the improvements we’ve made to the Android development workflow,” wrote Jamal Eason, product manager for Android, Google, in a blog post.



US launches open source repository, Code.gov

The US has finally taken a big step to promote the open source world and launched its Code.gov website. This move is aimed at offering the code of the federal government’s software to all the citizens. This comes hot on the heels of the release of the Federal Source Code Policy. The online repository already includes nearly 50 open source projects from over 10 agencies. This would grow over time. Also, the Barack Obama-led government is set to provide tools and support to agencies to implement its code policy.

“It is a step we took to help federal agencies avoid duplicative custom software purchases, and promote innovation and cross-agency collaboration. And it is a step we took to enable the brightest minds inside and outside of government to work together to ensure that federal code is reliable and effective,” wrote US chief information officer, Tony Scott, in a blog post.

The administration believes that Code.gov will become a ‘useful resource’ for government bodies as well as developers looking to build their offerings on the government’s code. This comes as an upgrade to the messaging bot which Obama launched last month.



Apart from the US, open source is influencing governments and authorities all across the globe. Last month, Russia showed it favoured open source software by reducing its dependence on US software vendors like Oracle, Microsoft and IBM. The Indian government is also in the process of launching a similar repository in the near future.

Google had originally announced the stopping of support and development for the ADT in Eclipse in 2015. However, the latest Android Studio release helped the company complete the awaited transition.

Android Studio 2.2.2 includes features like DDMS, Trace Viewer, Network Monitor and CPU monitor to offer developers a close alternative to the Eclipse tools. Additionally, the fresh Android Studio version comes preloaded with better accessibility such as keyboard navigation enhancements and screen reader support to enable people to develop Android apps easily.

Developers who would like to move their existing Eclipse ADT projects to Android Studio just need to download its updated version and then go to the built-in 'Import Project' menu option. Google has also opened its support to enable bug filings and feature requests from the developer community.

Enthusiasts and open source contributors can go to the project page to access Google's code for Android developments.

CoreOS launches Operators to extend Kubernetes with new capabilities

Linux distribution maker, CoreOS, has launched Operators as a new open source container management concept. This is designed to extend Kubernetes and simplify container management. The operating system is known for its capability to maintain open source projects for Linux containers.

Operators is not standalone software from CoreOS. Instead, it depends upon Google's Kubernetes. The development works as a micro service to help developers in breaking down a complex application structure into discrete pieces.



kubernetes

CoreOS, in a blog post.

In typical cases, the programmer has to first reduce the complex tasks on a whiteboard to view the project, and then manually locate IP addresses of the server and configure them on three different machines. Operators can automate this process and save the developers' time. The concept can reduce the effort involved in all the manual work with one declarative statement.

Operators can even eliminate the layer of complexity of heavy scripting in complex applications. It also makes it easy to enable periodical backups of the application's state and recover the previous state from the existing backups.

The CoreOS team has developed two open source Operators -- the etcd Operator and Prometheus Operator. While the former enables developers to create, manage and distribute etcd clusters, the latter provides a solution to use with the Prometheus tool to monitor Kubernetes resources.

Developers can access the code of the etcd and Prometheus Operators from their GitHub repositories. CoreOS is banking on the Kubernetes community's support for the new launch.

Future drones to get powered by open source

Red Cat Propware, a young company that builds software solutions for unmanned aerial vehicles, has launched its open source software and services for the drone market. The company has also established its new headquarters at Humacao in Puerto Rico to kickstart the development of community-backed solutions for advanced drones.

To introduce new features for drones, Red Cat Propware is actively working on building a strong open source community. The company is considering open source as an opportunity for the fast-growing drones market.



"Adoption of drones for commercial and competitive racing is exploding, pushing the limits of the software and features," said Jeff Thompson, founder and CEO of Red Cat Propware, in a statement.

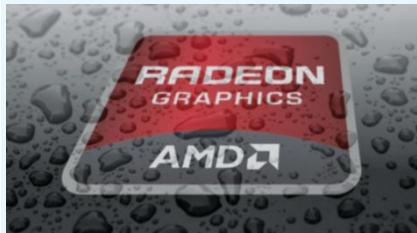
Founded in 2016, Red Cat Propware is offering not just software but also support and training in the drone market. The company also provides custom drone applications by leveraging the open source community's efforts.

Red Cat Propware is not the only company that prefers open source for the drone world. In August, Intel introduced its drone controller that offers an open source flight control platform to developers. Canonical also recently announced a development that uses the Ubuntu platform to transform drones into intelligent robots.

AMD expands graphics support for Ubuntu and Red Hat Enterprise Linux

AMD has released the AMDGPU-PRO 16.40 graphics driver with support for Ubuntu and Red Hat Enterprise Linux (RHEL). The updated driver has emerged over two months after the release of the previous graphics driver, and is built for various AMD Radeon R-series GPUs.

The latest AMDGPU-PRO version supports 64-bit Ubuntu 16.04 LTS as well as RHEL 7.2. It also includes support for APIs such as OpenGL 4.5 and GLX 1.4, OpenCL 1.2, Vulkan 1.0, VDPAU and Vulkan support for DOTA2. Additionally, there is an option to install script and Debian packages for Ubuntu 16.04.



In addition to its support to an expanding range of Linux operating systems, the AMDGPU-PRO 16.40 driver includes support for AMD Radeon R9 M485X, R7 M465, R7 M460, R7 M445 and R7 M440. The updated driver also comes with FirePro features such as EDID management and 30-bit colour support.

AMD has acknowledged some limitations alongside the upgraded support on the new graphics driver. It lags while producing graphics for the 'Company of Heroes 2' game and, on certain platforms, users are unable to log in to the system after its installation.

You can download the updated AMD graphics by visiting its global support website. The site also offers the same driver package for RHEL 6.8.

Maru OS now comes with Android Marshmallow

Maru OS, which got open sourced earlier this year, has now been updated to version 0.3. The new update brings Android 6.0.1 Marshmallow to all virtual environments, including desktops and mobile devices.

Originally running on Android 5.1 Lollipop, Maru OS now provides the Marshmallow flavour. The arrival of the new Android platform on Maru OS comes along with features such as improved power management, enhanced app permissions and up-to-date security patches.

Apart from Android

Marshmallow, Maru OS v0.3 allows users to start the Maru Desktop experience on a large screen even without an HDMI screen. Users just need to enable Maru Desktop from the dashboard to run the service in the background. This helps if the Maru OS-enabled phone is yet to be plugged into an HDMI display. Also, you can use SSH services if you have switched to the desktop mode.



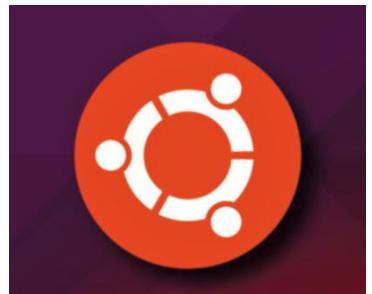
For high-resolution displays with over a 1080p pixel-count, Maru OS now comes with an 'Enhanced resolution matching' mode. This grabs the native resolution of the connected display and overrides the device's stock matching algorithm. The updated version additionally includes several performance improvements and bug fixes.

You can upgrade your existing Maru OS device to version 0.3 by following an upgrading guide on GitHub. It is worth noting here that if you are about to install the operating system for the first time and want to experience virtual environments on Android, you need to have a Nexus 5.

Ubuntu Core 16 brings security closer to IoT devices

Security has been one of the significant concerns in the open source world over the past several months. But now, Canonical has released Ubuntu Core 16 to ensure a reliable and secure experience on Internet of Things (IoT) devices.

The latest Ubuntu Core is a compact platform. Yet, it is capable of delivering what the company claims is groundbreaking security through confined, read-only Snap packages. The operating system also comes with Update Control to enable software publishers and manufacturers to validate updates way before they are applied to the devices. This helps in reducing instances of vulnerabilities.



To offer transactional upgradability for the entire platform, the operating system and kernel in Ubuntu Core are also delivered as Snaps. Manufacturers can use the device-centric Snap app store on Ubuntu's site to let developers release updates throughout the device's lifecycle, starting from beta testing to general availability.

IoT device makers like Dell believe that the release of Ubuntu Core 16 will enable them to offer long-term support and security on their offerings. This would help them influence more customers to test their innovations.

“Dell has been working with Canonical on Ubuntu Core for over a year, and our Dell Edge Gateways are fully-certified for Ubuntu Core 16. This enables Dell to offer the long-term support and security that IoT use cases such as factory and building automation demand,” said Jason Shepherd, director of strategy and partnerships for IoT, Dell, in a statement.

Linux has so far been the first choice for IoT device manufacturers. However, some serious issues emerged recently as warning against untested deployments. The Mirai botnet surfaced in October this year, and exposed thousands of connected devices to DDoS (distributed denial of service) attacks. Most recently, NyaDrop emerged, which loads malware on hardware such as DVRs and CCTV cameras.

Ubuntu Core is one of the popular solutions for devices ranging from top-of-the-rack switches and industrial gateways, to radio access networks, digital signages, robots and drones. Thus, its upgrade would bring enhancements onto a variety of devices and enable refined security across the entire IoT ecosystem.

Alfresco Activiti 1.5 comes with extensive data modelling

Alfresco Software has released the Alfresco Activiti 1.5 business process management (BPM) solution. The new update comes with extensive data

modelling features and offers one-click access to connected databases.

To improve Big Data developments, Alfresco Activiti 1.5 includes support for external data sources. Users can also

leverage the integrated enterprise content management (ECM) systems such as Alfresco One to fulfil their data requirements. Additionally, there is an option to ‘persist’ the changing data on their records and even keep a changelog for the underlying database.

“It is a basic tenet of our design philosophy that an inherently powerful, full-featured but complex application must be made easy to architect by non-professional developers. That is what we have accomplished with Alfresco Activiti 1.5 – allowing developers of all levels to tap content to enrich existing or new business processes,” said Paul Hampton, senior director of product marketing, Alfresco, in a statement.

The latest Alfresco Activiti includes integral content rule functions from Alfresco One to allow users to develop automated rules that will alter content under specific, pre-stated conditions. App developers can utilise content management features on the Alfresco platform by using the flexible design of the updated BPM tool. There is also a rich documentation functionality to let users document the content and flow of business processes.

Analysts believe that Alfresco Activity is a ‘rising star’ in the IT market, as it offers flexible design and many integration capabilities. “We are seeing accelerating interest among enterprises to automate as many processes as feasible – especially IT related processes, a trend that bodes well for both open source and vendor sponsored business process engines,” stated Carl Lehmann, principal analyst for enterprise architecture and process management at 451 Research.

Developers can access the advanced features of Alfresco Activiti 1.5 either on-premises or via a private cloud. It is available for a 30-day trial through the official site.



HTML 5.1 is now the latest Web standard

The World Wide Consortium (W3C) has released the official HTML 5.1 specification. The new release is the first minor revision of the fifth major version of the Hypertext Markup Language (HTML) that is being widely used across Web apps.

“In this version, new features continue to be introduced to help Web application authors, new elements continue to be introduced based on research into prevailing authoring practices, and special attention continues to be given to define clear conformance criteria for user agents in an effort to improve interoperability,” the W3C team wrote in a blog post.



Unlike its previous version that debuted in 2014, HTML 5.1 is not a big release. However, it brings some new attributes and elements such as `srcset`, `<picture>`, `<summary>` and `type="context"`. The newest revision also comes with the `requestAnimationFrame` API to enhance Web animation effects.

Alongside the new additions, the upgraded HTML standard includes tweaks such as nested `<header>` and `<footer>` elements, and the optional `url=` attribute. The consortium has removed some old features, like media controllers and command API.

W3C plans to bring out the HTML 5.2 recommendation sometime in late 2017. In the meantime, developers can start testing the features of HTML 5.1.



Are You In Suspense?

Are you wondering what happened to your order or payment, even after a fortnight? But did you mention your name and address clearly while sending payment, and what the payment was for?

We get several payments every month without adequate instructions or even identity of the person sending the payment. As a result we are unable to take appropriate action on these, till we get a complaint - if at all.

At EFY, we want you to be our satisfied customer. So please do let us know in case your payment or instructions are not even acknowledged within a fortnight. We are not that lazy!

Email: support@efy.in
Ph: 011-26810601 / 02 / 03 extn: 202

EFY GROUP
Technology Drives Us

Microsoft open sources hyperscale cloud hardware design

Expanding its verticals to retain market leadership, Microsoft has open sourced its next-generation hyperscale cloud hardware design. The new offering by the Redmond giant is a part of the Open Compute Project (OCP) that was jointly launched by Facebook, Google, Intel and Microsoft in 2014.

The design, called Project Olympus, is a new model for open source hardware brought out by the OCP community. It applies an open source collaboration model that has already been embraced for software, which is completely distinct from the current process for open source hardware developments.

“We are taking a very different approach by contributing our next generation cloud hardware designs when they are approximately 50 per cent complete – much earlier in the cycle than any previous OCP project,” wrote Kushagra Vaid, general manager, Azure Hardware Infrastructure, in a blog post.

Microsoft is set to enable the community to contribute to its ecosystem by downloading, modifying and forking the unfinished hardware design. This would work similar to open source software.

Experts believe that this will help in bringing advancements to the existing open source hardware development process. “Project Olympus, the re-imagined collaboration model and the way they are bringing it to market is unprecedented in the history of OCP and open source data centre hardware,” said Bill Carter, CTO of Open Compute Project Foundation.

The initial designs of Project Olympus include a new universal motherboard, a high-availability power supply and a battery. To fulfil global data centre needs, there is a 1U/2U server chassis, high-density storage expansion, universal rack power distribution unit (PDU) for global data centre interoperability and a standards-compliant rack management card. These modular blocks will be available independently, subject to requirements.

“We believe Project Olympus is the most modular and flexible cloud hardware design in the data centre industry. We intend for it to become the foundation for a broad ecosystem of compliant hardware products developed by the OCP community,” Vaid added.

Microsoft has released the motherboard and PDU specifications of the project on the OCP GitHub branch. Also, the entire rack system will soon be available as open source hardware.

Microsoft is not the lone player in the emerging world of open source hardware. Facebook is also actively developing its latest telecom and networking solutions for the community. These developments are predicted to get bigger over time.

Facebook develops open source networking infrastructure

Facebook has extended its Telecom Infra Project (TIP) and developed a new transponder platform called Voyager to deliver a scalable and cost-effective infrastructure solution. The new device is based on packet-optical technologies to enhance bandwidth delivery with cost-efficiency and customisability.



The very first version of Voyager is designed to leverage data centre technologies that were debuted with the top-of-the-rack switch, Wedge 100. It has a switch ASIC to aggregate the 100 GbE client signals. Additionally, there is the DSP ASIC and the optics module (AC400) from Acacia Communications to deliver an upgraded networking solution in the market.



Facebook's team has used the open line system that includes Yang software data models and an open northbound software interface to enable scalability on the new hardware infrastructure. The social networking giant has partnered with Snaproute for the software architecture of the end-to-end solution.

"An open approach allows any vendor to contribute new hardware and software to the system. In the beginning, the open line system will include Yang software data models of each component in the system, and an open northbound software interface (NETCONF and Thrift) to the control plane software," Facebook engineers Ilya Lyubomirsky, Brian Taylor and Hans-Juergen Wolfgang Schmidtke explained in a blog post.

Facebook has provided a hardware management daemon-based network element software stack with Voyager. The daemons enable configuration, management and monitoring of the hardware layer and provide services like higher-layer software to allow the provisioning of the hardware. Also, a multi-language SDK layer is available to enable third-party app development on the advanced infrastructure.

Facebook has already tested how Voyager operates in field trials with Equinix in the US and MTN in South Africa. The company is also aiming to release the code of the Voyager software to enhance its platform.

The development of an optical fibre-supported transponder like Voyager will certainly upgrade the present networking field. Besides, the open source approach will help Facebook to quickly grab attention from not just telecom operators but also several developers and data centre providers around the globe.

Ubuntu gets 'Budgie' flavour

Linux distribution Budgie-remix has transformed into a new Ubuntu flavour. With this development, the open source build is now available as 'Ubuntu Budgie'.

The team behind the platform confirms that the Ubuntu Developer Membership Board has passed Budgie as the official Ubuntu flavour after reviewing its technical aspects. For users, the transformation will bring community standards to the distribution.



"We have come a long way in a short time with our first 16.04 release — a major update at 16.04.1 as well as following and taking an active part with the Ubuntu release cadence for 16.10," the Budgie team wrote in the announcement statement.

Amazon Linux container image now available for on-premise data centres

Amazon has released the Amazon Linux container image for on-premise data centres. This new release enables Amazon Web Services (AWS) clients to deploy the same customised Linux experience on their own servers, which previously were limited to virtual machine instances by the e-commerce giant.



In addition to its on-premise presence, the Amazon Linux image can be deployed on the cloud. It is available through the EC2 Container Registry and built using the same code and packages that were initially available within the Amazon Linux AMI, which offers a 'stable, secure and high-performance' execution environment on AWS.

"Many of our customers have asked us to make this Linux image available for use on-premises, often as part of their development and testing workloads," AWS chief evangelist Jeff Barr wrote in a blog post.

The Amazon Linux image is not the only open source distribution available for AWS data centres. CentOS, CoreOS and even Red Hat Enterprise Linux and Canonical's Ubuntu are compatible with the on-premise data centres. However, the newest image is designed to use EC2 and limited remote access, with no root login and mandatory SSH key pairs to deliver a security profile. It also supports container solutions like Docker to enable advanced developments.

In addition to standards, there will be a new Ubuntu Budgie community to enhance the operating system. Developers will also get a chance to use help sites like Ask Ubuntu, Ubuntu Forums or Launchpad.net to easily ask for support on the latest platform.

Moving from just being another Linux distribution to an official Ubuntu flavour was not an easy task for the Budgie team. In fact, this massive task required several software changes, packaging updates, merging updates upstream and testing the results.

The official Ubuntu Budgie release will be available with the 17.04 release. There are also plans to add Budgie-desktop 11.

Ninety-eight per cent of developers use open source at work

Open source scales new heights each day. But a new study that surfaced online claims over 98 per cent of developers use open source tools at work.

Git repository manager GitLab has conducted a survey that revealed some interesting facts about open source adoption. The survey, conducted with a developer group, claimed that of the 98 per cent of developers who prefer open source usage at work, 91 per cent opt for the same development tools for work and personal projects. Moreover, 92 per cent of the group consider distributed version control systems (Git repositories) as crucial for their everyday work.



Among all the preferred programming languages, JavaScript comes out on top with 51 per cent of respondents. It is followed by Python, PHP, Java, Swift and Objective-C. Also, 86 per cent of developers feel security is a prime factor for judging the code.

“While process-driven development techniques have been successful in the past, developers are searching for a more natural evolution of software development that fosters collaboration and information-sharing across the life cycle of a project,” said Sid Sijbrandij, CEO and co-founder of GitLab, in a statement.

GitLab surveyed 362 startup and enterprise CTOs, developers and DevOps professionals who used its repository platform between July 6 and July 27, 2016.

Linux Foundation now manages the JavaScript community

The Linux Foundation has announced the transition of the original JQuery Foundation into the JS Foundation to support a vast variety of JavaScript projects. This new collaboration will help the JavaScript community under the new mentorship programme.



In addition to the Linux Foundation, the JS Foundation has founding members such as IBM, Ripple, Samsung, Sense Tecnic Systems, SitePen and the University of Westminster, among others. The objective of the new group is to ‘drive broad adoption’ as well as support the ongoing development of JavaScript solutions and ‘facilitate collaboration’ within the developer community.

“The JS Foundation aims to support a vast array of technologies that complement projects throughout the entire JavaScript ecosystem,” said Kris Borchers, executive director, JS Foundation, in a statement. “We welcome any projects, organisations or developers looking to help bolster the JavaScript community and inspire the next wave of growth for application development,” he added.

The JS Foundation is not only set to focus on mentoring projects on the client side but also on the server side. Target areas of the JavaScript-centric group will revolve around application libraries, mobile application testing frameworks, JavaScript engines and ecosystem technologies.

The list of the initial projects under the JS Foundation Mentorship includes the Appium testing automation framework, the JerryScript JavaScript engine, the Mocha testing framework, Moment.js date library and the Node-RED programming environment. These initiatives will now operate in a community-driven environment.

The Linux Foundation is set to develop an open and technical governance model that includes a technical advisory committee and a governing board with representatives from member organisations. The group will work with standards bodies like W3C, WHATWG and ECMA TC39. Additionally, the Node.js Foundation will work closely with the group to select various open source projects.



Anil Seth



Programming with Objects in Alice 3

In this article, the author, who likes to explore different kinds of software, introduces readers to Alice 3 and indicates how to go about installing it as well as create games and stories with it.

My first exposure to the Alice learning environment was when I listened to the remarkable Randy Pausch's Last Lecture, <http://www.cmu.edu/randslecture/>. Alice has been on my list of software to explore for years now, and I finally got around to doing that after exploring Scratch.

Like Scratch, there is a stage, or rather, a 3D world, in Alice. You populate your world with objects and then program the objects to do what you want. As with Scratch, it is an excellent way to tell stories or create interactive games. Alice is more complex than Scratch as there is one additional dimension, and a special object—the camera. You view the world through the camera, which you may also program.

You can download Alice 3 from alice.org. Note that it is almost 1.5 GB! The reason for the large download size is that it comes with a large gallery of artwork, which made it possible even for me, who has trouble drawing stick figures, to experiment with 3D animation.

A programmer's perspective

As a programmer, the first thing you may notice when you start Alice 3 is the keyword, *this!* The connection to Java is obvious.

When you set up the scene to add the objects you want, you will notice that you can browse the gallery by *Class Hierarchy*. The classes include:

- **Biped:** Adult, teen, alien, rakshasa, skeleton, etc
- **Flyer:** Chicken, peacock, penguin, etc
- **Swimmer:** Fish classes, marine mammal classes, etc

In the *Edit Code* mode, you can browse the procedures and functions associated with each object in your scene.

Chances are that a non-programmer may not even notice the above!

Getting started

A non-programmer is more likely to browse the gallery by themes, e.g., Africa, Amazon, fantasy, ocean, wonderland, etc. Choosing wonderland will allow you to create objects like Alice, Mad Hatter, Mushroom, and so on.

The best way to learn is to follow some video tutorials. You can find a set of these from Duke University, <https://goo.gl/nYWpY4>. The beginner's tutorial, Witch's Cauldron, consists of seven videos, each lasting only four to five minutes. It is an excellent way to acquire an understanding of how to use Alice 3.

You will learn to:

- Create a world and add objects to it
- Position, orient and resize an object
- Position and orient the camera
- Code the actions of various objects

Creating a simple game

The next tutorial to try is how to create a simple game—the UFO Alien Rescue game — at the ICE Distance Learning Site, <https://goo.gl/vlRKRF>.

Here, you will learn more about creating objects and changing their properties.

On the programming side, you will learn how to control the game flow using the arrow keys and the space bar to move the spaceship over the aliens, and how to pick up each one. The programming constructs needed are the usual Conditional Statements, Creating and Using Variables, Handling Events, and Creating Procedures and Adding Parameters.

Alice 3 makes it easy to program a game like this one with functions like *isCollidingWith* to test if a beam has located an alien. You can set the opacity of the beam so that the aliens continue to be visible even if surrounded by the beam. You can then use the *moveTo* method to move an alien to the UFO, and set the vehicle of the alien to the UFO so that both move together.

Telling a story

Entertainment will continue to be what people need, even if robots do all the work. Apart from sports and games, the other major entertainment for all of us is story-telling. Hence, the final tutorial you should try is creating a trailer for 'Finding Nemo' from an Alice 3 workshop, <https://goo.gl/BAWBQy>.

The story you will tell in this case is:

Create a 5-10 second animation that shows Marlin's (a small fish) frightened reaction on meeting a shark.

As *SharkEncounter.a3p* does not seem to be available for download, you can create a new project using the sea floor. Add Marlin (ClownFish), Dory (BlueTang), a shark and a treasure chest from the gallery of the Ocean theme. You can now follow the tutorial.

Considerable time and effort will be spent in setting up the scene. Programming is easy once the scene is set up. Animations rely upon the duration option to indicate how long a particular step should last. It is fun to play around with Alice 3 and you are now ready to create your own stories! **END** 

By: Dr Anil Seth

The author has earned the right to do what interests him. You can find him online at <http://sethanil.com>, <http://sethanil.blogspot.com>, and reach him via email at anil@sethanil.com.

Services To Assess, Manage, Customise And Deploy, All Under One Roof

Lyra Infosystems: Enabling IT evolution

Deploying open source technologies is not an easy task. But there are certain companies that make the deployment seem like a breeze, by delivering effective and efficient results – offering a range of services under one roof. Bengaluru based Lyra Infosystems is one such concern, which has a vision of becoming a globally recognised, innovative, dedicated and productive IT consultant firm. Specialised in professional services and consulting, the company offers widespread support and comprehensive consultation for all open source technologies and deployment. These state-of-the-art services have been offered to its clients since the last decade.

Lyra offers implementation tools, upgrades, security and vulnerability resolution services to clients. It is also specialised in DevOps & ARA, RSM, SCM and information management services. With superior industry experience, Lyra's team has been proficiently providing cutting-edge solutions covering all the activities around OSS (open source support) services. It provides security, training and consultation, as well as legal remediation and other services.

The road to success

Rohit Sharma established Lyra in early 2007. Prior to this venture, Sharma worked in companies like SDRC (India), ISI (Integrated Systems Inc), Wind River Systems and PixTel Communications. With more than two decades of sales, marketing, operations and management experience, he has also been part of a couple of startups as a founding member, and helped establish them as successful and stable organisations. At Lyra, Sharma is responsible for sales and operations.

Distinguishing itself from the competition

Leveraging the first-mover's advantage, Lyra enjoys technological leadership because of its distinctive services. It is not only the first in the segment but also presently the only such firm in the Indian market. Lyra now enjoys good brand recognition, having been through a long learning curve that has led to more secure and efficient means of delivering its services and solutions.

To accelerate and design services and solutions in an innovative way, Lyra has strengthened its R&D team. It has senior management teams across all verticals, including sales, marketing, legal and IT, that help to pursue strategic alliances with the pioneers of DevOps, SCM and RSM. Also, it has plans to reach overseas markets and grow in new market segments.

Challenges faced during the journey to success

Being the first-mover, Lyra inevitably faced several challenges in creating and marketing the new services and solutions it offered. Apart from some financial challenges, Lyra was confronted with some other hurdles like the need for gap fixing while introducing a new solution in the region. The company has spent years to build awareness about its revolutionised solutions.

Amongst other challenges, research emerged as one of the biggest for Lyra. Gathering primary and secondary data to back certain assumptions on business projections was the key behind the issue. The company also faced difficulties in finding reliable partners. To reap the maximum benefits out of a partnership, Lyra looked for organisations that were pioneers in their segment and had a good reputation amongst industry giants.

The benefits for clients

Lyra's distinctive services and solutions are used across the ASEAN region. Its clients have seen several operational and financial gains such as reduced overheads and optimum efficiency.

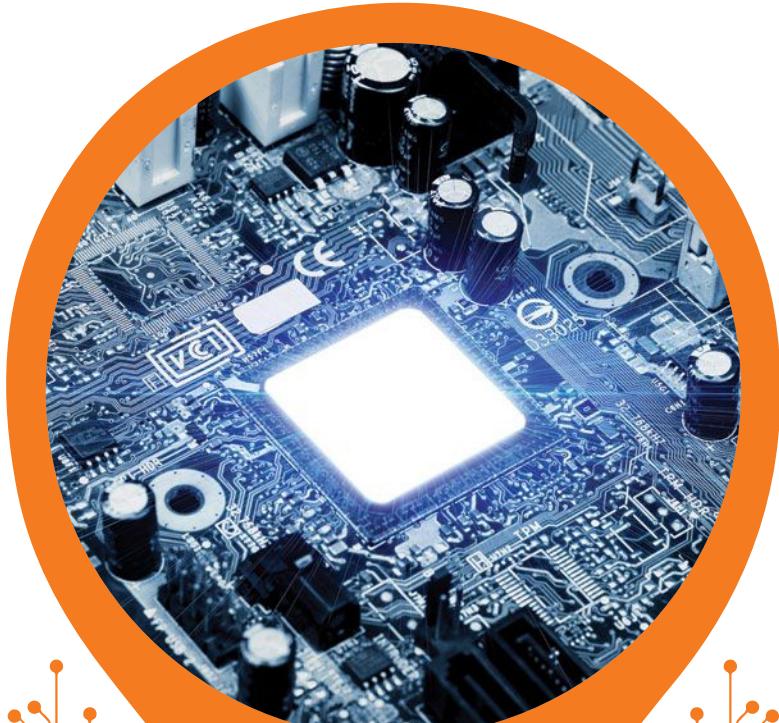
To reduce business risks, Lyra protects corporate IP and assists with compliance reporting. It also enables the implementation of a repeatable business process to support corporate compliance policies and gives deep insights into projects including known vulnerabilities, licence requirements and project activity. Additionally, it alerts companies when any new vulnerabilities are identified for their projects, and helps them manage and track remediation activities.

For enhanced security, Lyra adds an extra layer of online and network security. With its expertise in privileged access and password management, Lyra assists in incorporating privileged session management with a secure password vault that ensures privileged account passwords are protected.

Going forward with open source

An open source industry survey shows that 56 per cent of corporations contributed to open source projects a couple of years back. At present, the world is witnessing the next wave of open source. Companies like Twitter, Facebook, Netflix and Ericsson are participating in the OSS community, and developing and using open source in their own frameworks.

All in all, open source is the future of technology. Lyra's pioneering experience and expertise in the open source domain will definitely assist its clients with the adoption and correct usage of open source. 



Make sure
your software
solutions are
cutting-edge, not
cutting corners.



Lyra Infosystems is an industry leader in the comprehensive enabling of professional software solutions and practices, with best-in-class technologies and highly-trained professionals. Our areas of expertise range from, but are not limited to Open Source Support services, DevOps & ARA, SCM, and Remote Support Management. We ensure enhanced functionality, security, and efficiency while offering unparalleled customer satisfaction.

With businesses striving to gain a larger share of the market, there is a pressing need to stay ahead of the game. At Lyra, we provide solutions that translate into competitive advantage. With more and more businesses moving towards the open source system, we ensure that you are at the forefront of the next wave in enterprise software.



Open Source
Support Services



DevOps
& ARA



Legal
Remediation



Remote Support
Management



Software
Configuration
Management



Information
Management
Services



Consulting
services



Assessment
services

Develop and deploy with Lyra!

Lyra Infosystems Pvt. Ltd.

#149, 3 floor, 1 Block Koramangala, Bangalore - 560034, Karnataka, India

Email- mktg@lyrainfo.com

Open Source India 2016 Emerges as the One-stop Event for *Open Source Enthusiasts*

Open Source India (OSI) just hosted its 13th edition this year. The iconic convention had sessions by over 60 speakers. The 10 workshops held at this event were attended by a large number of open source supporters from India and around the globe. Apart from the expert talks, OSI also became a place for exhibitors to showcase their various community-centric offerings to hundreds of attendees.



The EFY Group hosted *Open Source India 2016* on October 21-22 at the NIMHANS Convention Centre in Bengaluru. The event was attended by various technology enthusiasts who were anxious to know more about what's new in the world of open source.

"The event was well planned in terms of the separate tracks for informative presentations and technical workshops," said Debabrata Nayak, project director—open source collaboration, National e-Governance Division.

Nayak was one of the prominent faces at this year's *Open Source India* and a key speaker, from a list that included Balaji Kesavaraj from Microsoft, Bruno Lowagie of iText Software, Andrew Aitken of Wipro and Virendra Gupta of Huawei India.

"There was a lot of interest among visitors for iText. Many people who attended my talk had some great questions and several high-quality candidates even applied for our openings in India," stated Lowagie, who is the founder of iText Software.

Attendees who were quite excited to learn about the latest open source developments at the event included developers, IT managers and CIOs, among others. Apart from the professionals, the convention was well attended by many evangelists and entrepreneurs.

"*Open Source India 2016* offered the right balance of content, which focused on sharing technical knowledge as well as the trends in open source adoption at the enterprise," said Monojit Basu, founder of TechYugaddi.

All trends under one roof

Open Source India 2016 featured talks on trending topics such as the cloud and Big Data, IoT, Web and mobile app development, and databases. It also kept audiences engaged with the inspiring back-to-back sessions at the Success Story track. CIOs and IT heads hosted some thought-provoking and engaging discussions throughout the two days of the event.

“We are glad to have presented DigiLocker as an open source success story from within the government, at such a platform. We are really overwhelmed and excited with the response of the open source community to our story, at the event,” Nayak told *Open Source For You*.

First-hand experience for techies

In addition to the technology sessions, the event featured hands-on workshops on open source hardware, software architecture, Web application security, Docker, DevOps and Data Lake as well as a whole-day OpenStack India conference. The convention also included a panel discussion on cyber security, dubbed as ‘Advanced cyber attack vectors using Web-based systems’.



Experts' panel discussing cyber security practices

“The cyber security panel discussion at *Open Source India* provided an opportunity for the audience to get some first-hand, real-time experience on how real hackers analyse and mitigate the APT risks,” said Divyanshu Verma, engineering manager, Ericsson R&D who hosted the discussion at a packed hall.

Companies and government bodies joined together

Leading technology companies such as Microsoft, Fiorano Software, iText Software, Wipro, Mafree, Huawei, Lyra Infosystems, Citrix, 2ndQuadrant, Siemens, Ashnik, Cloud Enabled, Asttecs and Zoho participated at the convention to support the growth of open source in the country. Also, the Indian government backed the initiative through the participation of its National e-Governance Division.

“We are delighted to have hosted another incredible edition of *Open Source India* this year. It was far bigger and more exciting than our expectations,” commented Ramesh Chopra, executive chairman, EFY Group.

Real-time updates from the convention were posted directly on Facebook and Twitter. Also, the inaugural sessions of the event were broadcast live to the world.

“We hope that the trend of supporting open source in India will continue in the future and that *Open Source India* will become the first choice for worldwide industry professionals as well as young developers and enthusiasts,” Chopra concluded.

POWERED BY

Platinum Partner



Silver Partners



FOSS For Everyone Partners



Associate partners



Lanyard Partner | Connectivity Partner



Organisers



Delegates praise the convention

Balaji Kesavaraj, director of platform strategy and marketing, Microsoft India

"Open Source India has become one of the most exciting gatherings for the open source world. This year's edition covered all the major technology trends, and Microsoft Azure was among the top hits at the convention. We saw that a huge number of attendees were keen to get their first impression of our open public cloud platform – Azure. They also got an opportunity to get hands-on, while working with open source technologies on Azure."

Tony Van den Zegel, director of sales, iText Software

"The organisation of the event was excellent. Also, the attendees showed lots of interest, came on time and stayed till the last sessions. I have witnessed different situations in the past, in India."

Ajay Bidari, VP, Cloud Enabled

"Indeed, it was great to be at Open Source India. We have received superb leads and exceptional support from the entire EFY team, which made us feel proud to be at such a highly organised edition."

Chanchal Bose, CTO, Prodevans Technologies

"Open Source India has emerged as a platform for people to communicate with experts who are already engaged in open source developments. It has become a great place to get training related to various software and hardware. Additionally, we got an opportunity to know what the computing world demands from a solutions provider like us. It has been a very exciting experience."

Dr Devasia Kurian, MD, Asttecs

"It is a great feeling to be here and quite interesting to see how the event has developed over the years. Today, several of the big guys are here—Microsoft, Citrix and some other tech giants who are interested in supporting open source. Also, the talks at the convention were done by some of the industry's prominent speakers. The event has also attracted companies that are into the proprietary business but are embracing open source. Overall, I would like to express my appreciation to the whole Open Source India team."

Sandeep Khuperkar, MD, Ashnik India

"It was a pleasure attending Open Source India. This is a platform where the community can meet the industry and enterprises, while the industry can understand better how technologies in the open source world can be used in the mainstream. Likewise, the industry feedback through the event helps the open source community to know the best ways to participate further, to bring these technologies into the mainstream. The event was also very well arranged, and sessions were well-timed; I did not see any delays. All in all, the EFY Group conducted the event quite well. I look forward to continuing to participate in Open Source India in the coming years."



Balaji Kesavaraj, director of platform strategy and marketing, Microsoft India, detailing

Suresh Govindachetty, lead sales engineer, Citrix Systems India

"The event was well organised and the people who attended the event evinced lots of interest in exploring the technologies we displayed. The support staff was helpful and guided us when needed."

Ankit Panchari, technology evangelist, Zoho

"This was the first time we participated in Open Source India. It was a wonderful experience to have interactions with the developer community here. The sessions were quite informative. Since we were aiming to reach the developer community through the convention, we met the right people. Overall, everything at the event was good."

Asheem Enoch Bakhtawar, regional director for India, 2ndQuadrant

"I really enjoyed this event, especially our lead generation, which went quite well. In fact, there were certain leads that I would never have gotten. It was very motivating for us to be able to reach out to the people whom we dreamt of meeting. At the same time, it gave us the opportunity to meet the open source community at one place. I am very grateful to EFY for hosting this convention."

Krishna M. Kumar, lead architect and delivery head for cloud and PaaS, Huawei Technologies

"I have been associated with Open Source India since the last three to four years. Thus, I can say that it has been improving every year. I felt the audience has grown and the quality of the sessions this year was much enhanced. There were several international speakers like Andrew Aitken and Bruno Lowagie, sharing their in-depth knowledge about the open source world."

Nityananda Panda, manager-training and consulting, Coss

Open source has now become the foundation of emerging technologies like OpenStack, Hadoop, Puppet, Docker and Ansible among others. Prodevans, our software solution and support wing and leading Red Hat solution partner, has also made its strong presence in the deployment and support world using open source and DevOps offerings. The Open Source India platform facilitates one-on-one interaction with the industry experts from the various field of specialisation and helps us keep in sync with emerging technologies. The event has emerged as an excellent opportunity to interact with the industry experts as well as the core community members at the same place.

Rahul Singh, marketing manager, Lyra Infosystems

"It was indeed a pleasure to be part of Open Source India. The event was professionally managed. Speakers and panellists were carefully chosen. The topics were very relevant and sessions were highly participative. The event was a great success for our team. We had the opportunity of meeting the extremely receptive (target) audience. The OSI team was very accommodating and helpful."

Key workshops @ Open Source India 2016

An introduction to Docker and the best practices to deploy it in production

The workshop made participants familiar with the Docker ecosystem. Neependra Khare, founder and principal consultant at CloudYuga Technologies, helped attendees to learn how Docker can be used for their primary work and to save time.

Software architecture: Principles, patterns, and practices

This workshop introduced participants to key topics in software architecture including architectural principles, constraints, non-functional requirements (NFRs), architectural styles and design patterns, viewpoints and perspectives, and architecture tools. Ganesh Samarthym, co-founder of CodeOps Technologies, presented examples and case studies from open source applications. The workshop also exposed attendees to some of the free or open source tools used by practising software architects.

Messaging based integration

This workshop focused on the ease-of-use of newly open sourced Fiorano ESB by demonstrating the benefits of messaging as the core of the ESB platform. Dhananjay Rao, a senior solutions architect at Fiorano Software, trained the attendees on product architecture, starting with an introduction to pre-built micro services and moving on to a live demonstration of Salesforce integration alongside some hands-on experience, followed by a Q&A session.

Building a data lake using Apache Hadoop: A proof-of-concept

In this workshop, participants were shown a proof-of-concept (PoC) of a data lake built on top of Apache Hadoop and Pentaho Data Integration, using data from financial markets. Monojit Basu, founder and director of TechYugadi IT Solutions and Consulting, answered questions about the importance of a data lake and how it can be built. The PoC he showcased enabled enthusiasts to learn about the open source platforms available for building a business data lake solution.

Security for Web applications

This workshop provided the attendees a platform to learn how to test a Web application using ZAP (Zed Attack Proxy). The objective was to find security vulnerabilities in Web

applications and help professionals like developers, penetration testers, security professionals, security consultants and IT managers with career-linked knowledge.

Building hackable keyboards with open source hardware and software

Through open source firmware, this workshop enabled attendees to get fresh insights into how free software and hardware can impact the development, programming and use of keyboards. Abhas Abhinav, founder of DeepRoot Linux, presented all the easy ways to customise keyboard layouts and shortcuts using open source solutions.

Security and DevOps

This workshop showed the relationship between security and DevOps, and answered the question on whether DevOps could kill information security. Vikas Prasar of Scalemonks highlighted all the major aspects of DevOps from a security expert's perspective.

Developing robust IoT applications for the enterprise

The prime objective of this workshop was to make the audience aware of the whole range of design factors involved in developing enterprise IoT applications. Debasis Das of ECD Zone revealed the

major issues developers face while building robust enterprise applications, detailing the design considerations and the current trends in the market.

Building embedded and IoT products with 96boards.org

This workshop helped developers prototype their embedded and IoT products using open source Web support through 96boards.org. Khasim Syed Mohammed of Linaro detailed the process of converting prototypes to end products with a reduction in BOM cost and of productising software.

A hands-on introduction to Docker

In this introductory hands-on session, Ganesh Samarthym of CodeOps Technologies introduced the concept of containers and provided an overview of Docker. Participants learnt the importance of Docker and the use of the Docker CLI (Command Line Interface) with basic features such as creating images and managing containers.



Open source learners practising new codes at a workshop

Key tracks@ Open Source India 2016

FOSS for everyone

This half-day track comprised sessions on free and open source software (FOSS). Experts from C-DAC, IBM, iText, Siemens and Unotech Software participated actively.

The cloud and Big Data

To lead participants into the advanced computing world, this track hosted topics such as hybrid cloud management trends and open source for the cloud and Big Data. Speakers from Ashnik, Citrix, Huawei and Microsoft participated.

Open source in IoT

This track featured experts from IBM, Microsoft, Dell and C-DAC, who highlighted open source's role in the IoT world. It included sessions on Apache Edgent, Bluemix Watson and the Azure IoT platforms.

IT infrastructure management

In this half-day track, participants were exposed to concepts like cloud DevOps, open source messaging and enterprise telecom building blocks. Speakers from Asttecs, Dell, Fiorano and Microsoft shared their knowledge.

Database management

This was one of the two full-day tracks at Open Source India.

It included speakers from 2ndQuadrant, Mafiree, Microsoft and Naukri.com, who hosted various sessions on the open source importance in database management.

Success stories

This full-day track detailed how open source has helped public and private organisations. Speakers from Airtel, Alef Mobitech, Jugnoo, Kesari Tours and TVS Motors participated, sharing their experiences. Also, the government's National e-Governance Division participated to reveal the DigiLocker journey.

Web app development

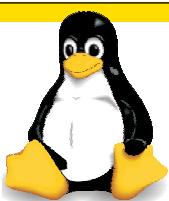
For the Web-first world, this included multiple sessions on micro services, Polyglot programming and API economy. Pundits from Cimpress, Hewlett-Packard Enterprise and Zoho shared their knowledge throughout the half-day track.

Mobile app development

This half-day session hosted various talks on the future of mobile apps. Representatives from Impetus Infotech, Microsoft and Twitter Communications spoke on trends like bots and hybrid applications.

By: Jagmeet Singh

The author is an assistant editor at OSFY.



Thank You Advisors & Speakers

13th Edition
OPEN
SOURCE INDIA
21-22
October
2016
NIMHANS Convention Center
BENGALURU

Advisory Panel



Dhiraj Khare
Business Manager,
EnterpriseDB



Divyanshu Verma
Engineering Manager,
Virtual Platform,
Ericsson R&D



Malati Kulkarni
Server Manager,
EMC Core
Technology



Janardan Revuru
Open Source
Evangelist,
Hewlett Packard
Enterprise



Balaji Kesavara,
Director of platform
strategy and
marketing,
Microsoft India





01: Rahul Chopra, editorial director, EFY Group

02: Atul Goel, vice president, EFY Group

03: Balaji Kesavaraj, director of platform strategy and marketing, Microsoft India

04: Queues for on-site registrations

05: Winner of Huawei Honor 8 contest

06: Bruno Lowagie, founder, iText Software

07: Andrew Aitken, global open source head, Wipro

08: Sandeep Khuperkar, managing director, Ashnik

09: Audience puts queries in front of speakers

10: Speakers at networking area

11: Shailesh Patel, managing director, Kesari Travels

12: Debabrata Nayak, project director—open source collaboration, National e-Governance Division

13: Attentive audience

14: Anant Kumar, GM and senior specialist - head product engineering, Bharti Airtel

15: Abhas Abhinav, founder, Deeproot Linux

16: Mohit Saxena, Sr. Software Engineer from Citrix talking on Linux Container.

17: Developers getting hands-on experience at Microsoft booth

Open Source India 2016



- 18: Ajay Bidari, VP, Cloud Enabled speaking at OpenStack India
 19: Vikash Jha, Co-Founder, Unotech Software, talking on Open Source and Real Time Data
 20: Ashok Sharma, CTO, QOS Technology conducted workshop on Securing Web Application
 21: Sagun Baijal from CDAC Mumbai talking on Meeting the IT challenges with Open Source
 22: Rajesh Jeyapaul, Cloud Solution Architect, IBM speaking on Apache Edgent and Bluemix Watson IoT platform - An open IoT solution to streaming Analytics
 23: Abhradip Mukherjee Lead solution consultant, Wipro Limited
 24: Raghvendra Singh Dikhit, Sr.Solutions Architect, Impetus Infotech Pvt Ltd speaking on Hybrid Mobile application
 25: Exhibitors addressing queries
 26: Ramakrishna Rama, Director, DELL India R&D, talking on Open Source in IoT Architecture

- 27: Vijay Venkatachalam, Principle Engineer, Citrix Systems
 28: Viendra Gupta, Senior Vice President Huawei India R&D, Huawei Technologies India Pvt. Ltd delivering Keynote
 29: Anupam Ghosh from Siemens Technology talking on FOSSology
 30: Neependra Khare, Founder and Principal Consultant, CloudYuga Technologies conducting workshop on Docker
 31: Krishna M Kumar and Sanil Kumar D from Huawei Technologies sharing The Story of BigData
 32: Atul Saini, CEO and CTO, Fiorano Software Inc
 33: Gandhali Samant, Sr. Technical Evangelist, Microsoft speaking on Azure IOT and the Open Source Stack
 34: Brijraj Singh, Senior Technical Evangelist, Microsoft speaking on NoSQL



- 35: Dr. Devasia Kurian, CEO, *astTECS talking on Enterprise Telecom
 36: Monojit Basu, Founder Director, TechYugadi IT Solutions & Consulting interacting during workshop on Data Lake
 37: Raveendra Reddy, Software Engineer, DELL sharing knowledge on Servers Management with Redfish
 38: Rajesh Sola, CDAC Pune enlightening audience on IoT Protocols
 39: Speakers after having talked on different topics at Open Stack Mini Conference
 40: Amardeep Vishwakarma, Associate Vice President - Engineering, Naukri.com speaking on MyTrend for MySQL
 41: Experts from Linaro.org conducting workshop on Building Embedded and IOT products
 42: Abhishek Dwivedi, Director Technology, Cypress speaking on Monolith to Microservices

- 43: Pradeep Chandru, CEO, Mafiree sharing his knowledge
 44: Janardan Revuru, Open Source Evangelist talking about Perks of being Polyglot Programmer
 45: Ganesh Samarthym, Co-Founder, CodeOps Technologies conducting workshop on Software Architecture
 46: Pavan Deolasee, PostgreSQL Consultant, 2ndQuadrant
 47: Saurabh Kirtani, Technology Evangelist, Microsoft sharing his knowledge on Developing bots with the Bot Framework and Cognitive Services
 48: Aman Alam, Developer Advocate - APAC, Twitter Communications Pvt Ltd enlightening audience on Automating your build process with Fastlane
 49: Vishal Gupta, CEO, Teknospire speaking role of Open Source in Technology start-up.
 50: Ankit Pansari, Evangelist, Zoho Creator ZOHO
 51: Sanjay Dhakar, VP of Engineering, Jugnoo speaking on how open source has played crucial role in Jugnoo's success



“
PEOPLE ARE
NOW EVEN
DOING MACHINE
LEARNING IN
JAVASCRIPT
”

JavaScript has already emerged as the backbone of the fast-growing world of the Web. But how is open source enabling JavaScript to power advanced Web developments? **Brendan Eich**, the *creator of JavaScript*, spoke to **Jagmeet Singh** of **OSFY** to reveal the secrets of the open source framework. Eich also talked of how he enhanced the Web with **Mozilla** and the work he is doing now at **Brave Software**.

Q What was the foremost aim of developing JavaScript?

I joined Netscape to ‘do Scheme’ in the browser. But on my first day at work, I learned that Sun and Netscape were working on a Java integration deal. So, with Marc Andreessen directly, and Bill Joy at Sun supporting me, I came up with a plan to make a dynamic language with Java (or the C family) syntax, which people who were not professional programmers could write directly in the HTML Web page source. We wanted a scripting language to complement Java, akin to Visual Basic for Visual C++ in Microsoft’s Windows platform. This would empower more people to start programming, by gluing components together with a little script in the page. The components were projected to be either built-in (the ‘DOM level 0’ which I implemented along with JavaScript for Netscape 2), or we hoped they could be written in Java by higher-priced programmers.

Q JavaScript is presently ubiquitous in the world of the Web. What is the reason behind its success?

There are three reasons that I see as critical and related. JavaScript itself is certainly the first and foremost reason behind its success. The second reason is its enough powerful basic features. And the third one is the ease in extending

the framework and patching (so-called ‘monkey-patching’) and ‘object detection’. All this enabled Web developers to compensate for version differences and even extend old or incompatible browsers to resemble newer or different ones.

Q How has the open source community helped in making JavaScript the star of the Web world?

Even before Mozilla or an open source implementation of JavaScript, I used all the early adopter support techniques and energy that I had acquired over the years in software, going back to Silicon Graphics. I helped developers find workarounds and reduce test cases. I answered questions promptly. Also, I groomed helpful second and third (virtual or even real) team-mates. Netscape eventually allocated and hired more people to work on JavaScript in late 1996. Before that I had important volunteer help.

Q Do you consider JavaScript as a dominant factor in new, emerging technologies like IoT and wearables?

Thanks to Node.js and the module ecosystem it spawned, JavaScript has moved strongly into servers and IoT devices. People are now even doing machine learning in JavaScript. Early hobbyist-level work such as Johnny Five endures and grows. I expect these trends will continue over time.

Q When can we see the much anticipated JavaScript 2.0? Could we say that for Web developers, this new version will be worth waiting for?

We had already kind of created what JavaScript 2 eventually became, which was ES4. It just had to be abandoned as that fourth edition, and its best parts carried forward into ES6, now called ES2015. The ECMA-262 standard has now moved to an yearly release cadence, so there's no need to await a 'big bang' 2.0 that might not ever have come out under such an 'all-or-nothing' approach. This more closely matches how software is developed and tested at scale.

Q A year back, there was an announcement on WebAssembly, which you had also called a 'game-changer'. Can you tell us about the status of the project and the support from browser vendors?

I helped announce WebAssembly, which is directly descended from asm.js, the statically typed subset of JavaScript that can be optimised to near-native code speed on par with Google's Portable Native Client (PNaCl) and other such predecessors. Since then, all four top browser vendors have moved forward with WebAssembly, and just the other week, three of them announced their plans to release WebAssembly support, which is likely to go live in the first quarter of 2017.

Q The ECMA TC39 committee that develops and standardises the ECMAScript (JavaScript) has recently become open to community contributions. Can organisations from India join and contribute towards language specifications by leveraging this latest development?

ECMA still requires people to agree to give up ownership of intellectual property according to its 'Code of Conduct in Patent Matters', which has language for ECMA members and non-members alike. Still, having worked in TC39 for over a decade since helping reform it once Firefox had restarted the browser market, I think it always helps to have a corporate or not-for-profit organisation that can join ECMA.

Q Years after introducing the world to JavaScript, you co-founded the Mozilla project. Why was there a need for launching Mozilla?

I had finished JavaScript 1.0 and standardised it with excellent colleagues from IBM, Microsoft, Sun and other companies, as ECMA-262 Edition 1. Netscape had finally invested in a JavaScript team, led by Clayton Lewis, with outstanding programmers who could carry on the work both in the engine I had rewritten (SpiderMonkey; the first engine was called Mocha), and in ECMA TC39. I was ready for something new.

Because Netscape was being driven out of business due to Microsoft's abuse of its Windows OS monopoly on personal computers, Netscape's executives and top engineers

 Even before Mozilla or an open source implementation of JavaScript, I used all the early adopter support techniques and energy that I had acquired over the years in software, going back to Silicon Graphics. 

were warm to the idea of releasing the Netscape code as open source. This was a daunting task (free the lizard!) and the first big commercial, closed-source conversion to open source.

Moreover, the chance to be the chief architect of the mozilla.org project—to work from the start on an 'escape pod' for the browser, was one I could not pass up. It is hard to remember how unlikely the later success of Firefox seemed. But even in 1998, some of us believed that Mozilla could bring back the browser market from its then-impending inevitable monopolisation by Internet Explorer.

Q The original Mozilla project brought in Mozilla Corporation as a for-profit arm to support the Mozilla Foundation. Is it easy to garner profits with open source developments?

Initially, mozilla.org had no legal structure. It was a 'virtual organisation' inside Netscape, soon bought by AOL. But in 2003, we spun Mozilla out from AOL as a non-profit, the Mozilla Foundation, with no for-profit subsidiary. After some promises from the US IRS were broken, we ended up creating the Mozilla Corporation as a wholly-owned subsidiary of the Mozilla Foundation.

I cannot say this model is easy to transfer to other open source projects. It is hard, perhaps even impossible, as Mozilla Firefox was *sui generis* (unique). What I have seen work for other open source projects ranges from commercial dual-licence to indie-developed or consulting-funded and to SaaS subscriptions. No one has cracked the code on how to make predictable money from open source. We do know that creating great software can create even greater value through ecosystem effects such as easier integration, community- and partner-built extensions and standards for interoperability.

Q How is the Indian developer base contributing to Mozilla's open source solutions?

I am no longer at Mozilla and cannot say much on that topic, since I am busy with Brave, which is doing well around the world, including in India. We already have several volunteer Brave Squads around the country. Some of them are from Lucknow, Hyderabad and even Jaipur. They are active in fixing bugs for Brave as well as in localisation efforts (adding Hindi and Bengali). Developers can get involved via the official Brave repository on GitHub or by joining us @BeBraveIndia on Twitter, as well as following the Facebook group.

Q Unlike its progress with a Web browser and email client, Mozilla had a brief stint in the operating system space with its Firefox OS. What reason do you ascribe to its failure in the market?

Firstly, although I was an executive sponsor of Firefox OS (codenamed B2G, at first) along with Mike Shaver, we did not get the CEO or other executives fully on board in 2011 when we started. While there was a bigger opportunity in the still-immature Android market, it took us till 2013 to really get to market and, at that point, the window of opportunity was closing due to Google's lead and scale advantages.

Second, even if we had started with full force in 2011, we might have failed, again because of Android's scale and lead-time. A mobile OS requires commodity-scale hardware, deep software integration and, in many parts of the world, a distribution system that has gatekeepers to be paid. It is therefore quite hard for anyone, never mind Mozilla, to do a 'third OS' now. Some say it's impossible, but of course, there are viable Android forks and mods in Asia.

Some day, perhaps, on a device as different from today's smartphones as the first iPhone was from feature phones, there will be a new OS. It certainly needs to be open source for best testing, recruiting and developer ecosystem effects.

Q After JavaScript and Mozilla, you recently expanded your presence in the Web world with Brave Software. Why did you feel the need for a separate Internet security company? All of the big browser vendors are more or less captured by today's ad-based ecosystem for funding free websites. Apple is the least captured as it walked away from ads as a business (twice, I believe), and so you have seen it add content blockers as an app-install model for extending Safari with ad blocking, starting last fall in iOS 9. But none of the big browsers will block third party trackers and ads, by default. Google would be cutting off its main revenue source if it blocked ads, by default.

Meanwhile, malware in ads (so-called 'malvertising') is coming in through third party ad exchanges. This is just the worst threat, among many—from slow page loads, to drained batteries due to the radio running too much when loading ad-tech scripts and ads or videos. The threats include users being bothered by 'retargeting', overt privacy concerns, and data theft that is legal and does not quite rise to the level of malware. These threats drive users to adopt ad-blockers, which, in the best cases, eliminate bad ads but in general, also deprive websites of revenue from ads.

Brave proposes to fix this broken ecosystem by blocking trackers and ads, by default. It offers users a wallet in the browser, with which to anonymously and effortlessly pay their top sites, and we hope (this is still in the future) even pay users properly for their attention.

We are working now with many small publishers and (soon) with a few large ones to find better ways than ads. Brave cuts out the middle players and proposes several ways for marketers and users to support publishers. Given all the threats and inefficiencies of online advertising and the downward

trends, we believe the Web is ready for a new set of standards for anonymous and private ads and micropayments.

Our long term value is not in how we innovate to create just some future standards, rather it is our users' trust in us putting them first, always giving them the same or even larger cut of any revenue based on their attention, and never holding our users' data in the clear off of each user's devices. All this means we never hold cleartext user browsing data on our or anyone else's servers.

Q How is the Brave browser different from Firefox, despite both being open source offerings?

Brave blocks third-party ads and trackers, by default, and works hard for user privacy in other ways (HTTPS everywhere activated, by default, and fingerprinting protection). Other browsers, including Firefox, do not block adverts by default, because they do not want to rock the boat.

Brave also differs by being based on Electron and (same as Chrome's engine) Chromium. This gives us the market-designed best engine with the most widely used extensions, but not any of the Google accounts or other service integrations, nor the front-end features of Chrome, which we do not want. Brave has its own user interface, built via Electron using React.js, HTML, CSS, SVG and images.

For users, particularly mobile users, the speed with which Brave loads Web pages directly translates into benefits. For instance, Brave Android users will see a 2x to 4x speed increase, which correlates with battery and data plan optimisation.

Q Lastly, how do you see the parallel journeys of open source and the Web?

Canadian science-fiction author Cory Doctorow in his recent talk on the darker view of things notes that while open source has won, the Web is in trouble due to closed source and worse threats such as DRM and the laws enforcing it. I believe, open source, which goes back a long way to when IBM released its Fortran, is inevitable in any broad, commercial-in-part ecosystem that has stable kernels across evolutionary time, transforming from UNIX to Linux, TCP/IP, HTML and JavaScript. These kernels and analogues of them in other layers of the ecosystem's 'stack' become the cost rather than profit centres, over time. The Web now has over three giant open source engines backed by multi-million LOC (lines of code) projects, namely Mozilla's Gecko, the Apple-led WebKit and the Google-led Chromium. Microsoft has even recently open-sourced its new ChakraCore JavaScript engine.

However, I see not so much a parallel as a symbiosis. The Web is big enough and commercialised now, so inevitably its evolutionary kernels will only be open-sourced for best peer-review effects such as development, QA and evangelism. Open source, in turn, symbiotically promotes open standards by providing reference implementations and A/B whitebox testing which is better than any closed-source approach that feeds into standards. 

CODE SPORT



In this month's column, we discuss a few computer science interview questions.

As we reach the end of 2016, we reflect briefly on some of the technical trends of this year. 2016 has been the year when AI, machine learning, IoT, and augmented virtual reality have become the buzz words within the tech community. It is also the year in which Big Data technology started stabilising and the hype started dying down as realistic implementations and adoption gathered momentum. On-the-fly, in-the-moment content experience has become the norm with Snapchat, Twitch, Periscope, Facebook Live Streaming, etc. Virtual augmented reality became a popular topic with the release of the Pokémon Go game. One question I frequently get asked, especially by our student readers, is, "What are the hot technologies I should read up and work on?" Well, there is no easy answer to that. I strongly believe that one should focus on gaining strong computer science fundamentals, first. Building a solid foundation on computer science basics such as data structures, algorithms and operating systems enables you to make your own choice on specialising in any area that is of interest to you, be it machine learning, AI, IoT or augmented reality. The reason I advocate that students get their fundamentals clear is because the specific areas that are considered 'hot' for a while can change over time, but each of these areas requires you to have sufficient knowledge in the basics of computer science, which hardly changes over time. Though this may sound trite, I still would advocate that our student readers make sure that they develop a solid foundation in algorithms, data structures and OSs. Then one can decide to develop expertise in a specific area, whether it is cloud computing, Big Data, IoT or machine learning.

Another question that I get asked frequently by student readers is which programming language they should develop expertise in. Again, there is no single right answer. My suggestion is that one should have

adequate proficiency in a systems programming language like C, a popular programming language like Python and any other languages that you need to do the job on hand! The programming language is nothing but a tool to achieve a particular task. Given the enormous self-help resources available on the Web today, especially Stack Exchange, it is not difficult to venture into a new programming language if your current task warrants it. In fact, the list of popular programming languages has changed little over the last one year, as you can see at <http://www.tiobe.com/tiobe-index/>. Java, C, C++ and Python still continue to rank among the top popular programming languages. One of the interesting things to note is the increasing popularity of the language 'Go', which is perhaps due to the interest in cloud programming — in particular, the Dockers containers that use Go.

As is our regular practice, we close this year with a medley of computer science interview questions. We focus on a wide set of topics including algorithms, operating systems and machine learning.

1. What is the difference between using containers and virtual machines for resource control? When would you prefer containers over virtual machines and vice versa?
2. You are given an array A of N integers and a number K. You need to find out how many times, if at all, K appears in array A. What would be the time complexity of your solution if (a) the given array A is unsorted, and (b) the given array A is sorted?
3. Given an array A of N characters, can you find out what the first character is in the array A that does not repeat itself?
4. What are virtual functions in C++? How are virtual functions implemented?
5. What are strongly typed languages and weakly typed languages? Can you give examples for each?

6. Given an arbitrarily long character string, you are asked to find out the most frequently occurring character and the least frequently occurring character. Write a C function to solve this problem, given a character string of size N . Now you are told that instead of being a fixed size character string, you are given a stream of characters that are coming dynamically as input. At any point in time, you should be able to print the most frequent and least frequent characters in the stream (note that if two or more characters have the same highest frequency, you can choose to print any of them as the most frequent characters). What is the time and space complexity of your algorithm?
 7. You are given two sentences S_1 and S_2 , each containing N and M words, respectively. You are now asked to convert S_1 into S_2 by either deleting a word or adding a new word. Modifying word1 to word2 can be considered as word deletion followed by word addition. Each insert operation has a cost of $+2$ and each delete operation has a cost of $+1$. Given two arbitrary sentences S_1 and S_2 , the edit distance of (S_1, S_2) is the minimum total cost of operations needed to convert S_1 into S_2 . You are asked to write a function to compute the edit distance. What is the time and space complexity of your function?
 8. You have written a program which takes N seconds on a single processor system. Now you have been told to reduce the execution time of the application. So you decided to parallelise the application and run it on an 8-processor system. However, you found that only 70 per cent of the application can be parallelised. What is the speed-up you can expect when you run the modified code on the 8-processor system?
 9. You have been asked to write code to reverse a singly linked list. You wrote a non-recursive version, which can do this task. However, the interviewer contends that your solution is not good and a recursive solution would be better than a non-recursive solution, in terms of time and space complexity. Do you agree with the interviewer? If yes, explain why? If not, can you explain the time and space complexity of non-recursive and recursive versions for reversing a linked list?
 10. A couple of months back, a popular news item doing the rounds in the tech social media space was about a candidate getting dropped by a Google technical recruiter, based on some questions for which he was expected to give a stereotypical answer. The post makes for hilarious reading at: <http://www.gwan.com/blog/20160405.html>. However, some of the questions in the post are pretty relevant in understanding the candidate's conceptual understanding (though it requires that the recruiters should also know computer science and not just read from a script sheet that they have been given). One of the questions is about the efficiency of sorting algorithms. When would you prefer Quicksort to Mergesort?
 11. You are given a document corpus consisting of more than a 100,000 documents, with each document containing at least 1000 words. Given a document D , you are asked to find all documents that are similar to D . How would you go about doing this?
 12. In machine learning, there are different classification methods such as Naïve Bayes, Support Vector Machines, Random Forests, etc. Given a particular data set and a binary classification problem, how would you decide which classifier to try on the data set first? Does your selection of the classifier depend on the characteristics of the data set? If your answer is 'Yes', explain why. If not, justify why it does not depend on the data set.
 13. What is meant by multi-task learning? Can you explain the benefits of multi-task learning?
 14. You are given two singly-linked lists A and B , containing integers. You are asked to find the union and intersection of the two linked lists and represent the results in (a) linked list format, and (b) in a set format. What is the time and space complexity of your solution?
 15. You are given an array A of N integers and a number K . Find a pair of integers in the array A such that the sum of the pair of integers is closest to K . What is the time complexity of your solution if (a) the given array A is unsorted, and (b) the given array A is sorted?
 16. In operating systems, what is the difference between a deadlock and livelock? How would you distinguish one from the other?
 17. In UNIX, can you explain the difference between fork and exec system calls?
 18. What is the halting problem? Can you sketch a quick proof for the halting problem? Why is the halting problem so important in computer science?
 19. Can you explain, conceptually, the difference between a mutex, semaphore and a barrier?
 20. What is an atomicity violation? Can you give a small coding example of it?
- If you have any favourite programming questions/software topics that you would like to discuss on this forum, please send them to me, along with your solutions and feedback, at sandyasm_AT_yahoo_DOT_com. Till we meet again next month, happy programming! 

By: Sandy Mannarswamy

The author is an expert in systems software and is currently working as a research scientist at Xerox India Research Centre. Her interests include compilers, programming languages, file systems and natural language processing. If you are preparing for systems software interviews, you may find it useful to visit Sandy's LinkedIn group *Computer Science Interview Training India* at <http://www.linkedin.com/groups?home=HYPERLINK%22http://www.linkedin.com/group%23home=&gid=2339182%22&HYPERLINK%22http://www.linkedin.com/groups?home=&gid=2339182%22gid=2339182>

Apache Spark: The Ultimate Panacea for the Big Data Era

Apache Spark is a data analysis engine based on Hadoop MapReduce, which helps in the quick processing of Big Data. It overcomes the limitations of Hadoop and is emerging as the most popular framework for analysis.



With the advent of new technologies, the data generated by various sources such as social media, Web logs, IoT, etc, is proliferating in petabytes. Traditional algorithms and storage systems aren't sophisticated enough to cope with this enormous volume of data. Hence, there is a need to address this problem efficiently.

Introducing the Apache Spark engine

Apache Spark is a cluster computing framework built on top of Apache Hadoop. It extends the MapReduce model and allows quick processing of large volumes of data significantly faster, as data persists in-memory. It has fault tolerance, data parallelism capabilities and supports many libraries such as GraphX (for graph processing), MLlib (for machine learning), etc. These features have led to Spark emerging as the most popular platform for Big Data analytics and it being used by the chief players in the tech industry like eBay, Amazon and Yahoo.

Spark was created in 2009 by Matei Zaharia at UC Berkeley's AMPLab as a lightning fast cluster computing

framework. In 2010, it was donated to the Apache Software Foundation under a BSD licence and has since been developed by contributors throughout the world. In November 2014, Zaharia's enterprise, Databricks, sorted a large dataset in record time by using the Spark engine. Spark 2.0.0 is the latest release, which came out on July 26, 2016.

Hadoop has been widely used due to its scalability, flexibility and the MapReduce model, but it is losing its popularity to Spark since the latter is 100x faster for in-memory computations and 10x faster for disk computations. Data is stored on disk in Hadoop but in Spark, it's stored in memory, which reduces the IO cost. Hadoop's MapReduce can only re-use the data by writing it to an external storage and fetching it when needed again. Iterative and interactive jobs need fast responses, but MapReduce isn't satisfactory due to its replication, disk IO and serialisation. Spark uses RDD (Resilient Distributed Dataset), which allows better fault tolerance than Hadoop, which uses replication. Though Spark is derived from Hadoop, it isn't a modified version of it. Hadoop is a method to implement Spark, which has its own cluster management system and can run in

Gift Yourself Knowledge

And get more gifts from us

OFFERS FOR YOU & ME



Portronics

YOGG Wrist Band

Your personal fitness and activity tracker



10W LED Bulb
Branded 10W LED bulb
with a two-year warranty

USB Car Charger
Three USB ports with
micro USB cable



**10W LED Bulb +
Three Night Lamps**
Set of one 10W LED bulb
with three 0.5W night lamps



ORDER FORM

Magazine	Duration	Cover Price (₹)	You Pay (₹)	You Save (₹)	Gifts For You & Me
Electronics For You	1 Year	720	625	95	<input type="checkbox"/> 10W LED Bulb
	2 Years	1440	1150	290	<input type="checkbox"/> USB Car Charger
	3 Years	2160	1725	435	<input type="checkbox"/> 10W LED Bulb+3 Night Lamps
	5 Years	3600	2880	720	<input type="checkbox"/> Yogg Smart Wristband

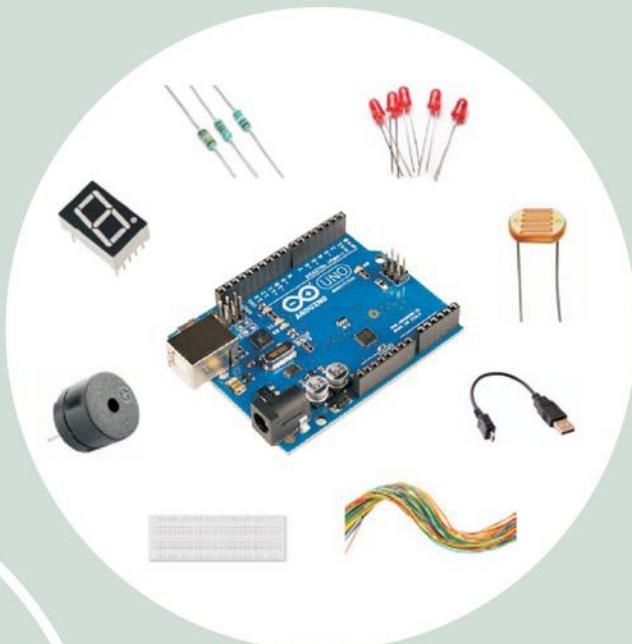
To Subscribe Online, visit:
<http://subscribe.efyindia.com>

**Technology Advisers for
Businesses and Techies**



SUBSCRIBE TO ELECTRONICS FOR YOU, SAVE MONEY & GET GIFTS

OFFERS FOR TECHIES



Arduino Starter Kit

A perfect kit to help you get started with Arduino projects.

Digital Multimeter

Use it to measure voltage, current, resistance and continuity of the circuit.



Basics Of Electronics Kit

A simple electronic kit which is packed with all the important components to help you learn by doing it yourself



IoT Development

A compact IoT embedded module for development of Internet based projects



ORDER FORM

Magazine	Duration	Cover Price (₹)	You Pay (₹)	You Save (₹)	Gifts For Techies
Electronics For You	1 Year	720	575	145	<input type="checkbox"/> Digital Multimeter
	2 Years	1440	1150	290	<input type="checkbox"/> Basic Of Electronics Kit/Components Kit
	3 Years	2160	1725	435	<input type="checkbox"/> IoT Development Board
	5 Years	3600	2800	720	<input type="checkbox"/> Arduino Starter Kit

Free ezine
Access With
Every Subscription



Name _____ Designation _____ Organisation _____

Mailing Address _____

City _____ Pin Code _____ State _____ Phone/Mobile _____ Email _____

Subscription No. (for existing subscribers only) _____ I would like to subscribe to Electronics For You 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)

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: 011-26810601-03 | Fax: 011-26817563 | e-mail: info@efy.in | www.efy.in **EFY GROUP** Technology Drives Us

Terms & conditions: # This offer is applicable for new subscriptions and renewal by existing subscribers. # The rates are valid for subscribers within India only. # Please allow 4-6 weeks for processing your subscription and 2 weeks to send your gifts # Replacement copies will be sent only if intimation of damaged / non-receipt of copies is received within 15 days of publication # If you are not satisfied with the magazine or our services, we shall refund the balance amount after three months. # Disputes, if any, are subject to exclusive jurisdiction of competent courts and forums in Delhi/New Delhi only.

standalone mode, hence obviating the necessity for the former. Hadoop provides only two functions to Spark—processing by MapReduce and storage using the Hadoop Distributed File System (HDFS). Spark doesn't replace Hadoop as the two aren't mutually exclusive. Instead, they complement each other and result in an extremely powerful model.

The power of the Apache Spark engine

Speed: Spark uses in-memory cluster processing, which means it reduces the I/O operations for iterative algorithms as it stores the intermediate data generated in the memory instead of writing it back to the disk. Data can be stored on the RAM of the server machine and, hence, runs 100x quicker in memory and up to 10x faster on disk as compared to Hadoop. Moreover, due to its bottom-up engineering and the usage of RDDs, the fundamental data structure of Spark allows transparent storage of data in memory and persistence to disk only when it's needed. 'Lazy evaluation' is a feature that also contributes to Spark's speed by delaying the evaluation of any expression or operation until the value is needed by another expression. This avoids repeated evaluation of the same expression, and allows the definition of control flow and potentially infinite sets.

Libraries: Spark is equipped with standard built-in high-level libraries, including compatibility with SQL queries (SparkSQL), machine learning (MLlib), and streaming data and graph processing (GraphX), in addition to the simple MapReduce functionalities of the MR model. These increase the productivity of developers by allowing them to use the functionalities in fewer lines of code, yet create complex workflows. Spark is compatible with real-time processing applications.

Multiple languages: Programmers have the advantage of coding in familiar languages as Spark provides stable APIs in Java, Scala, Python, R and SQL. The Spark SQL component allows the import of structured data and its integration with unstructured data from other sources. Spark has over 100 high-level operators as it is equipped with standard built-in high-level libraries, including compatibility with SQL queries (SparkSQL), machine learning (MLlib), streaming data and graph processing (GraphX) in addition to the simple MapReduce functionalities of the MR model. It can be used in real-time processing applications by applying transformations to semi-structured data with the option of allowing interactive querying within the Spark shell. This dynamic nature has led to it being more popular than Hadoop.

Hadoop support: Big Data and the cloud are synergistic and Spark's support for cloud technologies is one of its biggest advantages. It is compatible with widely used Big Data frameworks like HDFS, Apache Cassandra, Apache Hbase, Apache Mesos and Amazon S3. Spark, which doesn't have its own storage system, enhances the Hadoop stack by implementing it in three possible ways: 1) standalone mode, 2) over YARN, or 3) SIMR (Spark in MapReduce). It can also support existing pure Hadoop ecosystems.

MapReduce alternative: Spark can be used instead of MapReduce as it executes jobs in short, micro bursts of 5 seconds or less. It is a faster framework for batch processing and iterative algorithms in comparison to Hadoop-based frameworks like Twitter Storm for live processing.

Configuring Apache Spark on Ubuntu

It is easy to install and configure Apache Spark on Ubuntu. A native Linux system is preferred as it provides the best environment for deployment. Virtual OSs can also be used, but the performance gets compromised when compared to the native versions. Dual OSs work satisfactorily. There are options to use a standalone version or use a version pre-built for Hadoop, which utilises the existing Hadoop components such as HDFS or a version built to be deployed on YARN. The following section will explain how to get Spark 2.0.0 standalone mode running on Ubuntu 14.04 or later.

Installing Java: To install and configure Spark, your machine needs Java. Use the following commands in a terminal to automatically download and update Java:

```
$ sudo apt-add-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java7-installer
```

You can check for an existing version by typing:

```
$ java -version
```

Installing Scala: Spark is written in Scala; so we need it to install the former. Download version 2.10.4 or later from <http://www.scala-lang.org/>.

Untar the file by using the following command:

```
$ sudo tar xvf scala-2.10.4.tgz
```

Add an entry for Scala in the file `.bashrc`, as follows:

```
nano ~/.bashrc
```

At the end of the file, add the path given below to show the location of the Scala file:

```
export SCALA_HOME=<path-where-scala-file-is-located>
export PATH=$SCALA_HOME/bin:$PATH
```

Then we need to source the changed `.bashrc` file by using the command given below:

```
source ~/.bashrc
```

We can verify the Spark installation by using the following command:

```
$ scala -version
```

Installing Spark: Download the standalone cluster version of Spark from its website <http://spark.apache.org/downloads.html>

Then extract the file by typing the following command in the terminal:

```
$ tar xvf spark-2.0.0-bin-hadoop2.6.tgz
```

Add entry to `.bashrc` by:

```
nano ~/.bashrc
```

Add the line specifying the location to `~/.bashrc` by:

```
export SPARK_HOME=/home/sapna/spark-2.0.0-bin-hadoop2.6/
export PATH=$PATH:$SPARK_HOME/bin
```

Then source it by using the command below:

```
$ source ~/.bashrc
```

Start Spark services and the shell. Then let's change the directory by going into Spark's folder and manually starting the master cluster using the command shown below:

```
cd spark-2.0.0-bin-hadoop2.6
./sbin/start-master.sh
```

After running this, you can view the user interface of the master node by typing the following command in the browser:

```
http://localhost:8080
```

You can start the slave node by giving the following command:

```
./sbin/start-slave.sh <name of slave node to run>
```

To check if the nodes are running, execute the following:

```
jps
```

Architecture of the Apache Spark engine

Spark uses a master/worker architecture. There is a driver called the *Spark Context object*, which interacts with a single coordinator called the *master* that manages workers in which executors run.

Spark is founded on two chief concepts—the **RDD** (Resilient Distributed Dataset) and **DAG** (Directed Acyclic Graph) execution engine. An **RDD**, a read-only immutable collection of objects, is the basic data structure of Spark. The data is partitioned, and each **RDD** can be computed on a different node and can be written in many languages. It stores the state of the memory as an object across the jobs

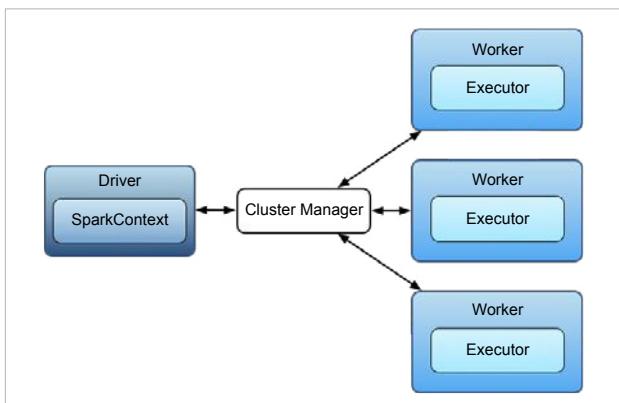


Figure 1: Architecture of the Spark engine

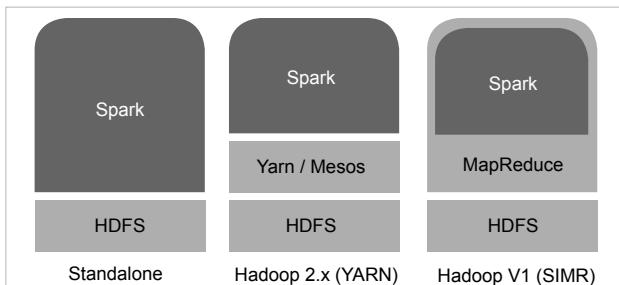


Figure 2: Possible deployment scenarios for the Spark engine

and the object is shareable between those jobs. **RDD** can transform data by mapping or filtering it, or it can perform operations and return values. **RDDs** can be parallelised and are intrinsically fault-tolerant. They can be created through two methods—by taking an existing collection in your driver application and parallelising it or by creating a reference from an external storage system like HDFS, HBase, AWS, etc. The **DAG** helps to obviate the multi-staged model of **MapReduce** which offers processing advantages.

Spark can be deployed in three popular ways to cater to different scenarios. The first way is to use a standalone mode. Here, Spark is placed above HDFS and allocates memory to it manually. All Spark jobs on the clusters are executed with Spark and MapReduce running simultaneously. The second way is to use a cluster management system such as Hadoop YARN (Yet Another Resource Manager), which doesn't require any pre-installation or root access to integrate with the Hadoop stack or ecosystem. Other components can be externally added on top to increase the functionality. The third way is to use SIMR (Spark in MapReduce) which, in addition to a manager, also executes a Spark job. Spark shell can be used without any administrative authorisation.

The main elements that constitute Spark are: Spark Core, MLlib, GraphX, Spark Streaming and Spark SQL. Spark Core is the basic platform engine that serves as a foundation for building other functionalities. The Spark SQL component, which provides the abstraction called SchemaRDD, which allows the loading, analysis and processing of semi-structured

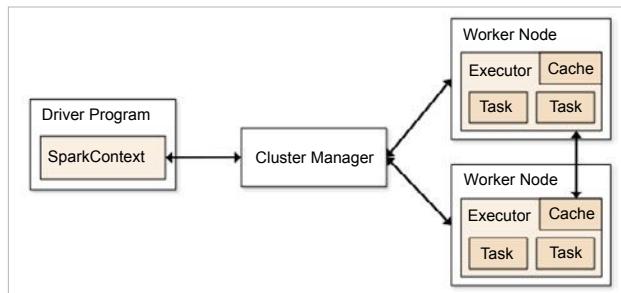


Figure 3: Internal architecture of the Spark engine

and structured datasets, is built on top of this. Spark Streaming allows live streaming and analysis of data loaded into RDDs as mini-batches. MLlib is an extensive library, which helps to implement machine learning methods on Big Data sets. It is created by a community of programmers from across the world. GraphX is a distributed graph-processing framework, which provides an API for expressing graph computation that can model the user-defined graphs by using the Pregel abstraction API. In addition, GraphX includes a growing collection of graph algorithms and builders to optimise graph analytics tasks.

Spark applications run independently on sets of clusters that are managed by a SparkContext object in the driver program. A SparkContext instance can connect with managers such as Mesos or YARN and can allocate resources to different commodity machines for optimised performance. After allocation, the executors on each job receive the application code and tasks, which are utilised to execute the job. Each Spark application has its own executors which can do multi-threading. Data needs to be stored on external storages for different Spark applications to share it.

Getting started with the Apache Spark engine

The following section explores how to start the Spark engine and get the services started. It will show how to execute existing programs, how to start the client or server and how to launch the shell.

Starting Spark services and the shell

We will change the directory, go into Spark's folder and manually start the master cluster by using the following command:

```
cd spark-2.0.0-bin-hadoop2.6
./sbin/start-master.sh
```

After running this, you can view the user interface of the master node by typing the following command in the browser:

```
http://localhost:8080
```

You can start the slave node by using the following command:

```
./sbin/start-slave.sh <name of slave node to run>
```

To check if nodes are running, execute the following:

```
jobs
```

Running the Spark shell

You can run the Spark shell for Scala using the command given below:

```
$ bin/spark-shell
```

You can run the Spark shell for Python by using the following command:

```
$ bin/pyspark
```

Submitting an existing application in Spark

First, let us compile a file that contains the code for a program which is to be run in Spark later on:

```
$ scalac -classpath "spark-core_2.10-2.0.0.jar:/usr/local/spark/lib/spark-assembly-2.0.0-hadoop2.6.0.jar" <file name>
```

Then, let's create a JAR file out of the compiled file, as follows:

```
jar -cvf wordcount.jar SparkWordCount*.class spark-core_2.10-1.3.0.jar/usr/local/spark/lib/spark-assembly-1.4.0-hadoop2.6.0.jar
```

Now, submit the JAR file to Spark to run the application, as follows:

```
$ spark-submit --class <application name> --master local <jar file name>
```

Writing and executing basic scripts in the Apache Spark engine

Since we have already learnt how to start the shell and submit jobs through it after creating and compiling JAR files, let's now write and execute a simple WordCount example in Scala to be deployed on Spark.

First, create a simple *input.txt* file from the sentence given below and put it in the Spark application folder containing all other jar files and program code:

"This is my first small word count program using Spark. I will use a simple MapReduce program made in Scala to count the frequency of each word."

Next, open the Spark shell:

```
$ spark-shell
```

Then make an RDD, which will read the data from our *input.txt* file. *sc* is SparkContext object, which is a manager

of all the RDDs:

```
scala> val inputfile = sc.textFile("input.txt")
```

We apply transformations to the data by splitting each line into individual words. Earlier, one line was one entity but now each word is an entity. Next, let's count the frequency of each word and then reduce it by its key, by adding the frequency of each distinct word, using the code shown below:

```
scala> val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_+_);
```

We can cache the output for it to persist, as follows:

```
scala> counts.cache()
```

Or we can store it to an external text file, as follows:

```
scala> counts.saveAsTextFile("output")
```

We can check the output as follows:

```
$ cd output/  
$ ls -1
```

Print the output on the Spark screen, using the command shown below:

```
$ cat part-00000  
$ cat part-00001
```

Analysing Big Data using the Apache Spark engine

With the advancement in technology, Web servers, machine log files, IoT, social media, user clicks, Web streaming, etc, are all generating petabytes of data, daily. Most of this is semi-structured or unstructured. This Big Data is characterised by high velocity, high volume and high variability; hence, traditional algorithms and processing technologies are unable to cope with it. MapReduce was able to process this data satisfactorily using a cluster of commodity hardware. But the ever-increasing volume of data is exceeding the capability of MapReduce due to the reasons mentioned earlier. Spark was designed as an answer to the limitations of MapReduce. It provides an abstraction of memory for sharing data and for in-memory computing. RDD can be persisted and re-used for other computations. Spark's multi-platform support, the ability to integrate with Hadoop, and its compatibility with the cloud make it tailor-made for Big Data.

In the real world, Spark is used for many applications. Banks analyse large volumes of data from sources like social media, email, complaint logs, call records, etc, to gain knowledge for credit risk assessment, customer

segmentation or targeted advertising. Even credit card fraud can be checked by it. E-commerce sites use the streaming clustering algorithm to analyse real-time transactions for advertising or to recommend products to customers by gaining insights from sources like review forums, comments, social media, etc. Shopify, Alibaba and eBay use these techniques. The healthcare sector benefits from Spark as it enables quick diagnosis and filters out individuals who are at risk. The MyFitnessPal app uses Spark to process the data of all its active users. Spark is widely used in genome sequencing and DNA analysis as millions of strands of chromosomes have to be matched. This task earlier took weeks but now takes only hours. Spark is also being used by the entertainment industry (such as Pinterest, Netflix and Yahoo News) for personalisation and recommendation systems.

Sample Big Data processing using the Apache Spark engine

Let's look at a simple application for beginners that can process Big Data. Let's load the dataset of 'Five Thirty Eight', a popular US TV show, and perform simple aggregation functions. Download the data for the past 50 years using https://github.com/fivethirtyeight/data/blob/master/daily-show-guests/daily_show_guests.csv.

Create an RDD, read the data and print the first five lines using the following code.

```
raw_data = sc.textFile("daily_show_guests.csv")  
raw_data.take(5)
```

Then, split each word by using a map function, as follows:

```
daily_show = raw_data.map(lambda line: line.split(','))  
daily_show.take(5)
```

Next, define a function to calculate the tally of guests each year, as shown below:

```
tally = dict()  
for line in daily_show:  
    year = line[0]  
    if year in tally.keys():  
        tally[year] = tally[year] + 1  
    else:  
        tally[year] = 1
```

Execute the function by using the Reduce transformation, as shown below:

```
tally = daily_show.map(lambda x: (x[0], 1))  
    .reduceByKey(lambda x,y: x+y)  
    .print()  
    tally.take(tally.count())
```

Now use a filter function, which segregates according to professions to create an RDD from an existing RDD:

```
def filter_year(line):
    if line[0] == 'YEAR':
        return False
    else:
        return True
filtered_daily_show = daily_show.filter(lambda line:
filter_year(line))
```

Now, execute this filter by doing reduce transformations:

```
filtered_daily_show.filter(lambda line: line[1] != '') \
    .map(lambda line: (line[1].lower(), 1)) \
    .reduceByKey(lambda x,y: x+y) \
    .take(5)
```

This completes the overview of one of the most promising technologies in the domain of Big Data. Spark's features and architecture give it an edge over prevailing frameworks such as Hadoop. Spark can be implemented on Hadoop, and its efficiency increases due to the use of both technologies synergistically. Due to its several integrations and adapters, Spark can be combined with other technologies as well. For example, we can use Spark, Kafka and Apache Cassandra

together — Kafka can be used for streaming the data, Spark for computation and Cassandra NoSQL database to store the result data. However, Spark is still being developed. It is comparatively a less mature ecosystem and there are a lot of areas, such as security and business integration tools, which need improvement. Nevertheless, Spark is here to stay for a long time. **END** 

References

- ```
[1] spark.apache.org/
[2] www.tutorialspoint.com/apache_spark/
[3] www.kdnuggets.com/2015/06/introduction-big-data-
 apache-spark.html
[4] https://www.infoq.com/articles/apache-spark-introduction
[5] https://jaceklaskowski.gitbooks.io/mastering-apache-
 spark/content/spark-architecture.html
[6] https://www.tutorialspoint.com/apache_spark/apache_
 spark_introduction.htm
[7] https://spark.apache.org/docs/0.8.0/cluster-overview.html
```

By: Preet Gandhi and Prof. Jitendra Bhatia

Preet Gandhi is an avid Big Data and data science enthusiast. You can contact her at [gandhipreet1995@gmail.com](mailto:gandhipreet1995@gmail.com).

Jitendra Bhatia has been working as a senior assistant professor with the CSE department of the Institute of Technology, Nirma University, for the past ten years. You can contact him at [jitendrabhatia@gmail.com](mailto:jitendrabhatia@gmail.com).

# 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](mailto: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

THE COMPLETE MAGAZINE ON OPEN SOURCE **For You**

# Varnish: A Performance Booster for Web Applications

Here's a superior caching engine for your Web applications. It will get them to work at blazing speeds with minimal configuration.



Web applications have evolved immensely and are capable of doing almost everything you would expect from a native desktop application. With this evolution, the amount of data and the accompanying need for processing has also increased. Apparently, a full-fledged Web application in a production set-up needs high end infrastructure and adds a lot of latency on the server side for processing repetitive jobs by different users. Varnish is a super-fast caching engine, which can reside in front of any Web server to cache these repeated requests and serve them instantly.

## Why Varnish?

Varnish has several advantages over other caching engines. It is lightweight, easy to set up, starts working immediately, works independently with any kind of backend Web server and is free to use (FreeBSD licence). Varnish is highly customisable, for which the Varnish Configuration Language (VCL) is used.

The advantages of this tool are:

- Lightweight, easy to set up, good documentation and forum support
- Zero downtime on configuration changes (always up)
- Works independently with any Web server and allows multi-site set up with a single Varnish instance
- Highly customisable with an easy configuration syntax
- Admin dashboard and other utilities for logging and performance evaluation

- Syntax testing and error detection of configuration without activation

There are only a few limitations to this tool. Varnish does not support the HTTPS protocol, but it can be configured as an HTTP reverse proxy using Pound for internal caching. Also, the syntax of VCL has been changing for various commonly used configurations with the newer versions of Varnish. It is recommended that users refer to the documentation for the exact version to avoid mistakes.

Varnish is a powerful tool and allows you to do a lot more. For instance, it can be used to give temporary 301 redirections or serve your site while the backend server is down for maintenance.

## Configuration and usage

Let us go through the steps to install and configure Varnish. For this tutorial, we'll use Ubuntu 14.04 LTS with the NGINX server.

Varnish 4.1 is the latest stable release, which is not available in Ubuntu's default repositories. Hence, we need to add the repository and install Varnish using the following commands:

```
apt-get install apt-transport-https
curl https://repo.varnish-cache.org/GPG-key.txt | apt-key add -
echo "deb https://repo.varnish-cache.org/ubuntu/ trusty
varnish-4.1" \
```

```
>> /etc/apt/sources.list.d/varnish-cache.list
apt-get update
apt-get install varnish
```

With this, Varnish is already running on your server and has started to cache. The Varnish configuration file is generally located at `/etc/varnish/default.vcl`.

Varnish has several built-in sub-routines, which are called the several stages of the caching fetch process. We can also define custom sub-routines, which can be called within these built-in sub-routines. The following are the built-in sub-routines for Varnish.

- `vlc_init` – Called every time before the configuration file is loaded
- `vlc_fini` – Called every time after the configuration file is discarded
- `vlc_recv` – Called when a new HTTP request is received by Varnish
- `vlc_pipe` – Called when the request is sent to the backend
- `vlc_pass` – Called before delivery, when the request is coming from the backend fetch
- `vlc_hit` – Called when the request is found in the Varnish cache
- `vlc_miss` – Called when the request is not found in the cache before forwarding to the backend
- `vlc_hash` – Called after the hash is created for the received request
- `vlc_purge` – Called when the cache is purged for the request
- `vlc_deliver` – Called when the output is delivered by Varnish
- `vlc_backend_fetch` – Called before fetching a request from the backend
- `vlc_backend_response` – Called after the response from the backend is received by Varnish
- `vlc_backend_error` – Called when fetching from backend fails after `max_retries` attempts

These sub-routines can be used in the VCL configuration file to perform the desired actions at various stages. This gives us high flexibility for customisation in Varnish. We can also check the syntactical correctness of the configuration file using the following command:

```
varnishd -C -f /etc/varnish/default.vcl
```

Varnish gives a detailed description of any error in the syntax, similar to what is available with NGINX and Apache servers.

## Performance and benchmarking

To see the actual difference in performance, we have used the Apache Benchmark tool, which is available with the `apache2-utils` package. For our tests, we have hosted a

```
ubuntu@ubuntu:~$ ab -n 100 -c 10 http://mywebsite.com/
This is ApacheBench, Version 2.3 (Revision: 3529065)
Copyright 1996 Adam Twiss, Zeus Technology Ltd. http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking mywebsite.com (be patient).....done

Server Software: Varnish
Server Hostname: mywebsite.com
Server Port: 80

Document Path: /
Document Length: 187782 bytes

Concurrency Level: 10
Time taken for tests: 8.344 seconds
Complete requests: 100
Failed requests: 0
Total transferred: 18778200 bytes
HTML transferred: 18778200 bytes
Requests per second: 294.85 [#/sec] (mean)
Time per request: 34.381 [ms] (mean)
Time per request: 3.438 [ms] (mean, across all concurrent requests)
Transfer rate: 53458.23 [Kbytes/sec] received

Connection Times (ms)
 min mean[+/-sd] median max
Connect: 0 8.7 1 3
Processing: 4 34.3 2.9 34
Waiting: 0 25 78.3 2 248
Total: 5 34 69.8 11 244

Percentage of the requests served within a certain time (ms)
 50% 11
 66% 13
 75% 14
 80% 15
 85% 26
 90% 242
 95% 242
 99% 244
 100% 244 (longest request)
ubuntu@ubuntu:~$
```

Figure 1: Benchmarking with Varnish

```
ubuntu@ubuntu:~$ ab -n 100 -c 10 http://mywebsite.com/
This is ApacheBench, Version 2.3 (Revision: 3529065)
Copyright 1996 Adam Twiss, Zeus Technology Ltd. http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking mywebsite.com (be patient).....done

Server Software: nginx
Server Hostname: mywebsite.com
Server Port: 80

Document Path: /
Document Length: 187782 bytes

Concurrency Level: 10
Time taken for tests: 5.000 seconds
Complete requests: 100
Failed requests: 0
Total transferred: 18778200 bytes
HTML transferred: 18778200 bytes
Requests per second: 379.99 [#/sec] (mean)
Time per request: 50.000 [ms] (mean)
Time per request: 5.000 [ms] (mean, across all concurrent requests)
Transfer rate: 3160.34 [Kbytes/sec] received

Connection Times (ms)
 min mean[+/-sd] median max
Connect: 0 54 1 4
Processing: 428 548 86.3 542 958
Waiting: 423 542 86.2 538 954
Total: 436 549 86.2 547 959

Percentage of the requests served within a certain time (ms)
 50% 542
 66% 576
 75% 591
 80% 593
 85% 623
 90% 737
 95% 958
 99% 959
 100% 959 (longest request)
ubuntu@ubuntu:~$
```

Figure 2: Benchmarking with NGINX (without Varnish)

fully loaded WordPress site on a `t2.micro` instance of EC2 in AWS. Let's call it `mywebsite.com` in our local host file to avoid DNS resolution delays in our tests. This server runs Varnish on Port 80 and the NGINX server on Port 8080.

The syntax to run the test is `ab -n <num_requests> -c <concurrency> <addr>:<port><path>`

Figures 1 and 2 show the statistics on running 100 requests with 10 concurrent threads.

Considering the benchmarking result, we get the mean time spent per request with Varnish as 3.438 ms, and without Varnish as 57.999 ms. It is evident that Varnish is a winner and a must-have tool for your Web servers, allowing you to boost the performance up to 1000x, depending on your configurations and architecture. 

By: Krishna Modi

The author has a B. Tech degree in computer engineering from NMIMS University, Mumbai and an M. Tech in cloud computing from VIT University, Chennai. He has rich and varied experience at various reputed IT organisations in India. He can be reached at [krish512@hotmail.com](mailto:krish512@hotmail.com).

# REST API Development Using Django Tastypie Framework

Tastypie is a Web service API framework for Django. It is used to create REST style APIs, which can be used from any application, and it comes with the serialisation and authentication standard that can be used while developing REST APIs. This article covers the development of a REST API using Tastypie.



**T**astypie is a powerful package for developing the REST API. It provides a convenient and powerful abstraction for developing the REST-style interface, which is very easy to implement. It also provides serialisation for XML and JSON. In this article, we will use JSON to return data.

For a better understanding of what this article covers, readers should be familiar with the basics of Django and the REST concept. In this article, I am going to cover the basics of Tastypie, such as resources, filtering and REST methods (GET, POST, PATCH, DELETE).

We will develop basic REST APIs for book management with a SQLite database and, along the way, we will learn to implement the REST API using Tastypie.

I am using Ubuntu 16.04 LTS as my development machine.

## Installation

First, let's install the following packages.

1. *Virtualenv:*

```
sudo apt-get install virtualenv
```

2. We will set up our project in *virtualenv*.
3. Create *virtualenv* and install the necessary packages in it:

```
virtualenv env
```

4. Activate *virtualenv*:

```
source env/bin/activate
```

5. Now, install the Django Tastypie package using Pip, as follows:

```
pip install django django-tastypie
```

## Project set-up

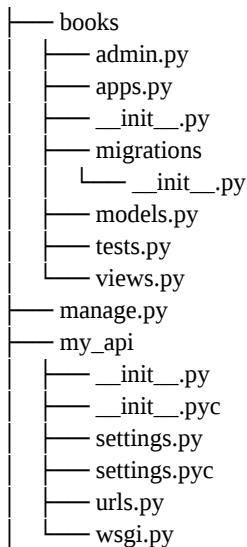
1. We will create a Django project called *my\_api*, with the following command:

```
django-admin startproject my_api
```

2. Switch to the *my\_api* directory and create an app called *books*:

```
./manage.py startapp books
```

3. After these steps, our working directory should look like what's shown below:



Let's start working on the code now.

## Creating models

Let's create two models for the *books* app, as follows, in *books/models.py*:

```
class Publication(models.Model):
 """Model for Publication
 """
 name = models.CharField(max_length=20)
 isbn = models.CharField(max_length=30)
 website = models.URLField()
 def __unicode__(self):
 return self.name
class Book(models.Model):
 """Model for Book
 """
 name = models.CharField(max_length=30, help_text="Name of the Book")
 author = models.CharField(max_length=30)
 description = models.TextField()
 price = models.IntegerField()
```

```
publication = models.ForeignKey(Publication)
def __unicode__(self):
 return self.name
```

## Configuration of Tastypie

We only need to add *tastypie* in *INSTALLED\_APPS* in *settings.py*.

```
INSTALLED_APPS = [
 'django.contrib.admin',
 'django.contrib.auth',
 'django.contrib.contenttypes',
 'django.contrib.sessions',
 'django.contrib.messages',
 'django.contrib.staticfiles',
 'books',
 'tastypie',
]
```

## Creating resources

For implementing REST APIs, 'resources' needs to be created, and to integrate Tastypie we need to create resource classes. We will create resource classes for each model. So, create the *books/api.py* file, and add resource classes within that file.

```
from tastypie.resources import ModelResource
from books.models import Book, Publication

class PublicationResource(ModelResource):
 """
 Resource for Publication
 """
 class Meta:
 queryset = Publication.objects.all()
 resource_name = 'publications'

class BookResource(ModelResource):
 """
 Resource for Book
 """
 class Meta:
 queryset = Book.objects.all()
 resource_name = 'books'
```

As *BookResource* and *PublicationResource* are subclasses of *ModelResource*, it will take all non-relational fields of models and maps to its own *ApiFields*, much like Django *ModelForms*.

In the above class, the *resource\_name* is optional. If not provided, it will automatically be generated from the class name, removing any instance of the 'Resource' and lower-casing the string. So 'BookResource' will become only 'book'.

## Hooking up the resources

Once we have the resource classes, we can hook the resources in *urls.py* so that it generates API end points for each resource. Let's create *urls.py* in our *books* app, as follows:

```
from django.conf.urls import url, include
from tastypie.api import Api
from books.api import BookResource, PublicationResource
api = Api(api_name='v1')
api.register(PublicationResource())
api.register(BookResource())
urlpatterns = [
 url(r'^api/', include(api.urls)),
]
```

Include *books/urls.py* in *my\_api/urls.py* as follows:

```
urlpatterns = [
 url(r'^admin/', admin.site.urls),
 url(r'^', include('books.urls')),
]
```

Once we are done with these steps, we are ready to play with Tastypie and fire up the REST API for books and publications.

Before running the server, let's first run the *migrations* command, which will sync models in the DB.

```
./manage.py makemigrations
./manage.py migrate
./manage.py runserver
```

## Using REST API URLs

Open another terminal and run a *Curl* command to hit API URLs.

```
curl http://localhost:8000/api/v1/books/?format=json
```

It will return an output as follows:

```
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 0}, "objects": []}
```

In this output, we have an empty 'objects' list because we don't have any books yet. So let's use the POST method to add book records.

## Creating new resources

Before adding a book resource, let's first add one publication resource, which will be attached to the book, as the Book model has the *publication* field as a foreign key to the Publication model.

```
curl -H "Content-Type: application/json" -X POST -d '{"name": "HarperCollins", "isbn": "123-456", "website": "https://harpercollins.co.in/"}
```

```
curl http://localhost:8000/api/v1/publications/ -i http://localhost
```

The above commands result in the following output:

```
HTTP/1.0 401 Unauthorized
Date: Sun, 09 Oct 2016 13:10:02 GMT
Server: WSGIServer/0.1 Python/2.7.12
X-Frame-Options: SAMEORIGIN
Content-Type: text/html; charset=utf-8
```

Oops! We are getting 401 status because, for reasons linked to safety, Tastypie ships with an *authorisation* class that is set to *ReadOnlyAuthorization*. This makes it safe to expose the API on the Web and to prevent anyone from doing POST/PUT/DELETE. Let's enable this in our resources, as follows:

```
from tastypie.authorization import Authorization
from tastypie.resources import ModelResource
from books.models import Book, Publication
class PublicationResource(ModelResource):
 """
 Resource for Publication
 """
 class Meta:
 queryset = Publication.objects.all()
 resource_name = 'publications'
 authorization = Authorization()
class BookResource(ModelResource):
 """
 Resource for Book
 """
 class Meta:
 queryset = Book.objects.all()
 resource_name = 'books'
 authorization = Authorization()
```

### Warning

Enabling authorisation is great for testing, but it should only be used during development, and the API should not be exposed to the Web with such resources. I have given a link for authorisation/authentication in the *References* section. Please spend some time in understanding those concepts and learn how to integrate them in Tastypie after reading this article.

So, let's now add a publication resource using POST:

```
curl -H "Content-Type: application/json" -X POST -d '{"name": "HarperCollins", "isbn": "123-456", "website": "https://harpercollins.co.in/"}
```

What follows is the output resulting from the above command:

```
HTTP/1.0 201 Created
```

```
Date: Sun, 09 Oct 2016 14:45:18 GMT
Server: WSGIServer/0.1 Python/2.7.10
Vary: Accept
X-Frame-Options: SAMEORIGIN
Content-Type: text/html; charset=utf-8
Location: /api/publications/1/
```

Cool...we got the 201 HTTP status code, which means that the publication information that we provided has been added to the DB. So let's get publication resources:

```
curl http://localhost:8000/api/v1/publications/?format=json
```

The above command will return publication resources as follows:

```
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects": [{"id": 1, "isbn": "123-456", "name": "HarperCollins", "resource_uri": "/api/v1/publications/1/", "website": "https://harpercollins.co.in/"}]}
```

In this JSON, “meta” is the meta information about the records returned; “limit” is the number of records to return in one API call -- we can configure the limit in *settings.py*; “next” and “previous” contain the URLs for fetching the next 20 records and previous 20 records, but as we have only one record, they are null; “total\_count” is the total record available for the resources; and, finally, “objects” is the dictionary list, which contains a list of the recorded information.

Tastypie does not handle representation of foreign keys by default, so we need to instruct it how to represent the foreign key field by adding the *ForeignKey* field to *Resource*. Configure *BookResource* to include *publication* as the foreign key field.

```
from tastypie import fields

class BookResource(ModelResource):
 """
 Resource for Book
 """

 publication = fields.ForeignKey(PublicationResource,
 'publication')

 class Meta:
 queryset = Book.objects.all()
 resource_name = 'books'
 authorization = Authorization()
```

The first parameter to *fields.ForeignKey* is the related field model resource class, and the second parameter is the related field name of the model (i.e., *publication* is the foreign key field of the Book model).

Now create the Book resource using POST, as follows:

```
curl -H "Content-Type: application/json" -X POST -d
```

```
'{"name": "The Alchemist", "author": "Paulo Coelho", "description": "The Alchemist is a novel by Brazilian author Paulo Coelho which was first published in 1988.", "price": 350, "publication": "/api/v1/publications/1/"}' -i http://localhost:8000/api/v1/books/
```

Notice that in the above command, the value for ‘publication’ key is the related resource detail’s URL; so book information is stored in the DB with *publication\_id* as 1.

Let’s get the book resources, as follows:

```
curl http://localhost:8000/api/v1/books/
```

```
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects": [{"author": "Paulo Coelho", "description": "The Alchemist is a novel by Brazilian author Paulo Coelho which was first published in 1988.", "id": 1, "name": "The Alchemist", "price": 350, "publication": "/api/v1/publications/1/", "resource_uri": "/api/v1/books/1/"}]}
```

To get the specific book resource, use ‘resource\_uri’.

```
curl http://localhost:8000/api/v1/books/1/
```

```
{"author": "Paulo Coelho", "description": "The Alchemist is a novel by Brazilian author Paulo Coelho which was first published in 1988.", "id": 1, "name": "The Alchemist", "price": 350, "publication": "/api/v1/publications/1/", "resource_uri": "/api/v1/books/1/"}
```

## Filtering

Let’s add some more book information and the names of the book publishers.

```
curl -H "Content-Type: application/json" -X POST -d '{"name": "George Newnes", "isbn": "0192123092", "website": "https://www.arthur-conan-doyle.com"}' -i http://localhost:8000/api/v1/publications/
```

```
curl -H "Content-Type: application/json" -X POST -d '{"name": "The Memoirs of Sherlock Holmes", "author": "Arthur Conan Doyle", "description": "The great detective series", "price": 450, "publication": "/api/v1/publications/2/"}' -i http://localhost:8000/api/v1/books/
```

```
curl -H "Content-Type: application/json" -X POST -d '{"name": "Westland Press", "isbn": "123-678", "website": "www.westlandbooks.in"}' -i http://localhost:8000/api/v1/publications/
```

```
curl -H "Content-Type: application/json" -X POST -d '{"name": "The Immortals of Meluha", "author": "Amish Tripathi", "description": "The shiva trilogy 1", "price": 350, "publication": "/api/v1/publications/3/"}' -i http://localhost:8000/api/v1/books/
```

```
400, "publication": "/api/v1/publications/3/" } -i http://localhost:8000/api/v1/books/
```

To use filters, add fields in resources to allow filters on those fields:

```
class PublicationResource(ModelResource):
 """
 Resource for Publication
 """

 class Meta:
 queryset = Publication.objects.all()
 resource_name = 'publications'
 authorization = Authorization()
 filtering = {
 'name': ['exact', 'contains']
 }

class BookResource(ModelResource):
 """
 Resource for Book
 """

 publication = fields.ForeignKey(PublicationResource,
 'publication')
 class Meta:
 queryset = Book.objects.all()
 resource_name = 'books'
 authorization = Authorization()
 filtering = {
 'publication': ALL_WITH_RELATIONS,
 'price': ['exact', 'lt', 'lte', 'gte', 'gt'],
 }
```

Now filter based on price

```
Curl http://localhost:8000/api/v1/books/?price=450
```

This will return the books object with a price= ₹450.

```
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects": [{"author": "Arthur Conan Doyle", "description": "The great detective series", "id": 2, "name": "The Memoirs of Sherlock Holmes", "price": 450, "publication": "/api/v1/publications/1/", "resource_uri": "/api/v1/books/2/"}]}
```

If we want all the books priced below ₹ 450, then use the “lt” keyword as follows:

```
curl http://localhost:8000/api/v1/books/?price_lt=450
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 2}, "objects": [{"author": "Paulo Coelho", "description": "The Alchemist is a novel by Brazilian author Paulo Coelho which was first published in 1988.", "id": 1, "name": "The Alchemist", "price": 350, "publication": "/api/v1/publications/1/", "resource_uri": "/api/v1/books/1"}, {"author": "Amish Tripathi",
```

```
"description": "The shiva trilogy 1", "id": 3, "name": "The Immortals of Meluha", "price": 400, "publication": "/api/v1/publications/1/", "resource_uri": "/api/v1/books/3/"}]}
```

Filter based on the publication’s name is done the same way that we filter for the name field by adding it to resource filters.

## Updating resources

To update the resource, we just need to provide fields to be updated for a specific resource.

```
curl -H "Content-Type: application/json" -X PATCH -d
'{"name": "The Nagas"}' -i http://localhost:8000/api/v1/books/3/
```

Now get the resource using the GET method. Then the output should contain the updated books’ names.

## Deleting resources

To delete the particular resource, use the DELETE method of REST.

```
curl -H "Content-Type: application/json" -X DELETE -i http://localhost:8000/api/v1/books/3/
```

Now, when you GET the book with ID 3, the 404 status should be returned.

```
curl -H -i http://localhost:8000/api/v1/books/3/
```

To summarise, Tastypie is a great framework for implementing the REST API in Django. A major advantage is the great community that stands behind it, resulting in good documentation and it being well tested. In this article, we looked at the basic CRUD operation on resources, and we can now implement custom authentication and authorisation, override the method of resource classes, and implement our own method as per our requirements. 

## References

- [1] <http://django-tastypie.readthedocs.io/en/latest/index.html>
- [2] <http://django-tastypie.readthedocs.io/en/latest/authentication.html>
- [3] <http://django-tastypie.readthedocs.io/en/latest/authorization.html>
- [4] <https://docs.djangoproject.com/en/1.10/>

## By: Yogesh Kamble

The author is currently working at Synerzip Softech Pvt Ltd as senior software developer. He is interested in working on back-end-technologies and loves to try out new Linux distros. He can be contacted at [yogesh.kamble@aol.com](mailto:yogesh.kamble@aol.com)



## Create Your Own Java Based Chat Robot

If you are interested in creating your own chat robot, then here is a tutorial on how to do it yourself by using Java and AIML.

**A** chat robot or chatterbot is a human chat simulator. It is a program for auditory or textual conversation between a computer and a human being. Such robots are used for fun, education and 24x7 customer services. When students and customers have many questions to ask, the robot gives the answers on behalf of teachers or customer service executives.

In this article, I have used AIML, A.L.I.C.E, Java and NetBeans. A brief introduction to these technologies follows.

AIML (Artificial Intelligent Markup Language) is an XML based mark-up language to help create a chat robot from scratch. It was first developed by Dr Richard Wallace when he created a chat robot named A.L.I.C.E (Artificial Linguistic Internet Computer Entity).

The important tags for AIML are listed below.

- **<aiml>**: The parent tag to start and end the AIML document.
- **<category>**: Every new question or pattern with its relevant answer goes in this tag.
- **<pattern>**: Matches the pattern with the user's question.
- **<template>**: The robot gives the answer from the

template if the pattern can be matched.

```
<xml version = "1.0" encoding = "UTF-8" ?>
<aiml version = "1.0.1" encoding = "UTF-8"?>
<category>
 <pattern> What is your name? </pattern>
 <template>
 I am Alice, nice to meet you.
 </template>
</category>
</aiml>
```

Here is an explanation of the code given above. File *first.aiml* contains the XML based tag for the robot's knowledge. Here, the **<category>** tag is used to describe the user pattern or the user's question. **<template>** is the response given by the robot to the user if the user's pattern is matched.

A.L.I.C.E is the robot created in 1995 by Dr Richard Wallace, in Java, by using AIML. We can also say that A.L.I.C.E is an AIML parser. Nowadays, many parsers are available in various languages like PHP, Python, etc.

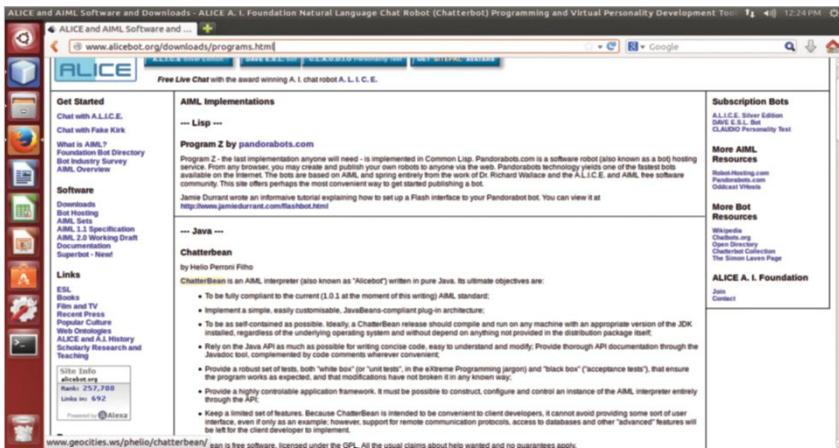


Figure 1: Downloading Chatterbean

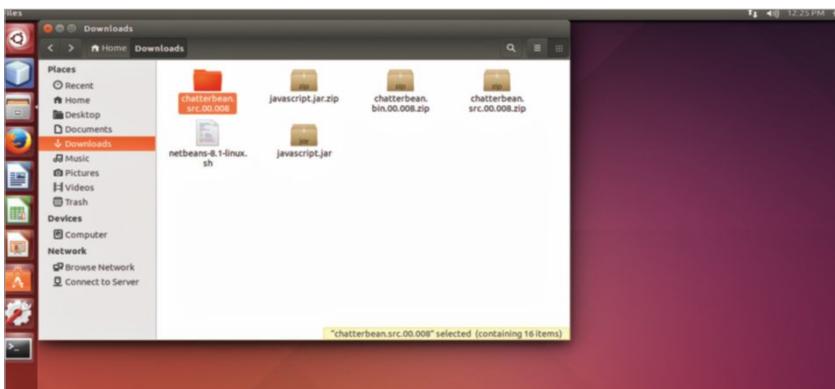


Figure 2: Extracting Chatterbean to the folder

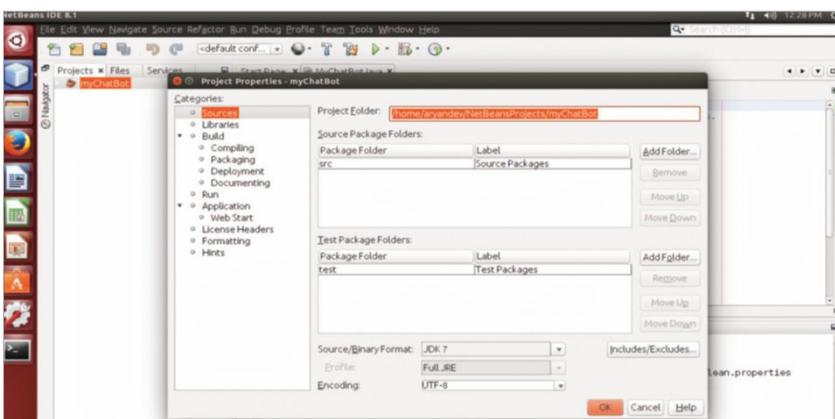


Figure 3: Creating a project and copying its path



Figure 4: Copying 'botoflife' to NetbeansProject/mychatbot/src

## Steps to creating a chat robot

1. Download the source code from <http://www.alicebot.org/downloads/programs.html>
2. Go to the Java section.
3. Download the source code under the link *Chatterbean*.
4. Under the section 'Download, Building and Usage Information', go to 'Download Chatterbean 00.008 Source Distribution'. We will download the source code rather than the binary distribution.
5. Extract Chatterbean to the folder (Figure 2).

This extracted folder contains the 'Bot' directory, which will have the collection of AIML. We can create our own AIML to increase the knowledge of our robot. But first let's proceed to the set-up. The folder 'Source' contains Java code for the robot A.L.I.C.E and the AIML parser. We will use this, as it is, in our program.

## Preparing NetBeans for coding

The following steps prepare NetBeans for coding.

1. Create a project named 'myChatBot'.
2. Copy the existing source code of A.L.I.C.E into our *Project* folder. Copy *botoflife* from the downloaded source code to *NetbeansProject/mychatbot/src* as shown in Figure 4.

The source code will appear in the *Projects* tab under *Source Packages* at the NetBeans IDE shown in Figure 5.

3. Now copy the *Bot* folder from the downloaded source code to *NetbeansProject/myChatBot* as shown in Figure 6.
4. Add additional supportive libraries which come with the A.L.I.C.E source code. These are *bsh.jar* and *junit.jar*.

The set-up is now ready. Open *MyChatBot.java* which contains *main()* and import the libraries, using

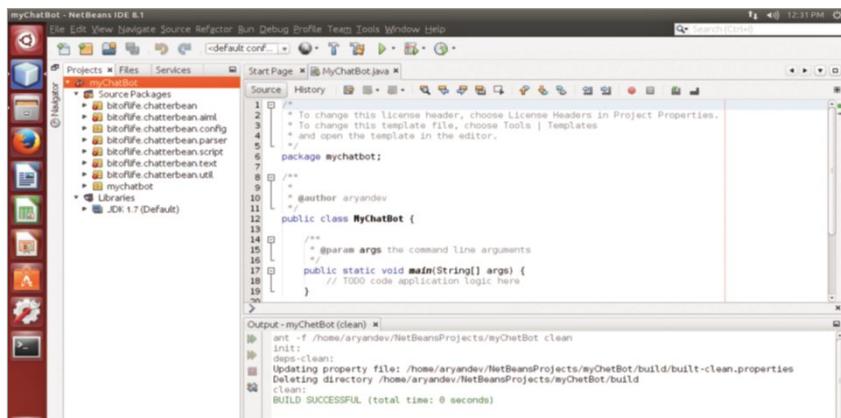


Figure 5: Package added in Project



Figure 6: Copying the Bot folder in the Project folder

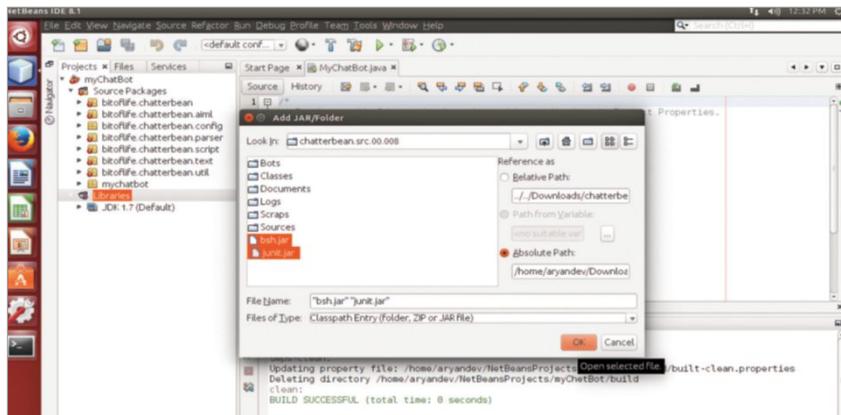


Figure 7: Adding the required libraries to Project

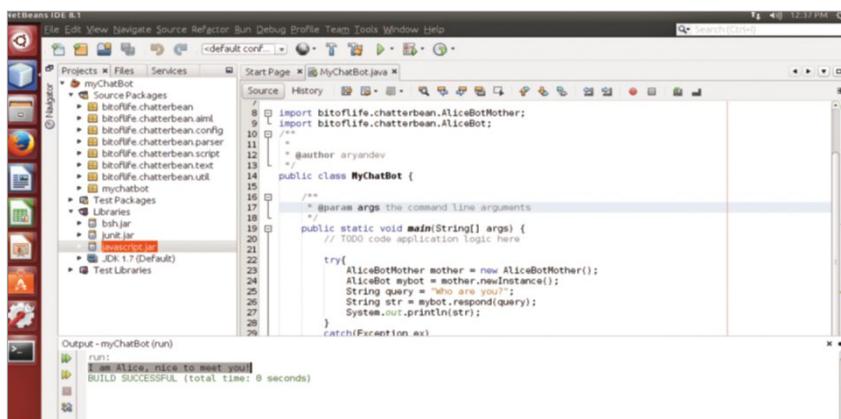


Figure 8: Final run: The answer given by A.L.I.C.E to the question

the following code:

```
import bitoflife.chatterbean.
AliceBotMother;
import bitoflife.chatterbean.AliceBot;

public static void main(String[] str)
{
 try{
 AliceBotMother mother = new
 AliceBotMother();
 AliceBot mybot = mother.
newInstance();
 String ask = "Who are you?";
 //Here You can ask Dynamic question.
 String str = mybot.
respond(ask);
 System.out.println(str);
 }
 catch(Exception ex)
 {
 System.err.println(ex.
toString());
 }
}
```

Execute the code. It will generate the answer shown in Figure 8.

In the above code, in the first two lines, the *AliceBotMother* and *AliceBot* class create the instance of Bot. Now, ask a meaningful question; in our case, it's *String ask*. Now pass this string to object *AliceBot* with the *respond* function, which returns the string answer predefined in the relevant AIML. We can ask various questions by replacing the string in *ask* variable.

Your own, personal robot is now ready! As it's a Java based application, you can use it on various platforms.**END** 

## References

- [1] <http://www.alicebot.org/downloads/programs.html>
- [2] <http://www.alicebot.org/aiml.html>

## By: Jignesh Prajapati

The author is assistant professor at Vadodara Institute of Engineering Research, an enthusiastic blogger and also has his own website named [drjslab.org](http://drjslab.org). He can be contacted at [j.prajapati@drjslab.org](mailto:j.prajapati@drjslab.org).

# Creating a Barcode Generator in App Inventor 2

This tutorial will help you create a barcode generator app using App Inventor 2. It will be beneficial for readers joining now to refer to earlier issues to gain mastery over App Inventor 2.



1234500067890

**B**arcodes are quite commonplace in daily life. Because of things going digital nowadays, you see the check-out counter staff in shopping complexes and malls scanning the barcode pasted on the product rather than manually entering the data about a particular item. A barcode is an image that has bars of varying thickness and spaces, and retains information within that pattern. Different barcode images will have different information written into them. We have handheld barcode readers or smartphones that decode the information written in a barcode. A barcode can have information in any form like, names, numbers, addresses, the price, quantity, etc. The more the information in a barcode, the more dense will be the arrangement of bars in the image.

In this article, we will make a barcode generator application using App Inventor 2. By now, readers will have gained sufficient experience and hands-on practice with this tool. If this is the first time you are reading an article in this series, you can very well learn from this point itself, without any prior

programming or specialised knowledge. Apps developed with App Inventor can be uploaded to Google Play.

We can proudly say that we have mastered App Inventor in terms of the components that are available for use in the palette. And playing with the designer and block editor is a lot of fun.

## Theme of the application

The theme is pretty simple and you probably have already got the idea from previous articles. We will make an Android application which will convert any written text to the barcode. The barcode generated can be saved to the SD or can be shared with other users via sharing components.

## GUI requirement

For every application we have a graphical user interface or GUI, which helps the user to interact with the on-screen components. How each component responds to user actions is defined in the block editor section.

## GUI requirements for Screen 1

1. **Label:** Labels are static text components that are used to display

some headings or markings on the screen.

2. **Button:** A button will let you trigger the event and is a very essential component.
  3. **Horizontal Arrangement:** This is a special component that keeps all the child components horizontally aligned within themselves.
  4. **Notifier:** This is used to display some instructions or give controls over your existing components. You will be able to explore its functionality in more detail as we implement it in our game.
  5. **Canvas:** This is the component for drawing and animation. You can place various animation objects over Canvas to control them via user actions. We will look at more details as we work with the application.
  6. **Text Box:** This is the component to take user inputs. Whatever text the user writes in the text box will be treated as input and can be used further. We will see more of it within the application itself.
- We will require the components listed in the table on the next page for this application. We will drag them on

|   | Component's name       | Purpose                          | Location                                   |
|---|------------------------|----------------------------------|--------------------------------------------|
| 1 | Label                  | To display a label               | Palette-->User Interface-->Label           |
| 2 | Button                 | To trigger events                | Palette-->User Interface-->Button          |
| 3 | Horizontal Arrangement | To arrange the child components  | Palette-->Layout-->Horizontal Arrangement  |
| 4 | Notifier               | To display on-screen information | Palette-->User Interface-->Notifier        |
| 5 | Canvas                 | To enable drawing                | Palette--> Drawing and Animation--> Canvas |
| 6 | Text Box               | To take written input from users | Palette-->User Interface-->Text Box        |
| 7 | Sharing                | To share the generated code      | Palette-->Social-->Sharing                 |

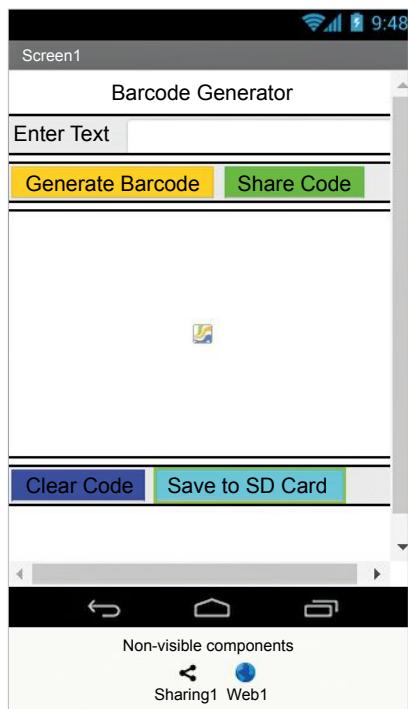


Figure 1: Designer screen

to the designer from the left hand side palette.

1. Drag and drop the components mentioned in the table above, to the viewer.
2. Visible components can be seen by you while the non-visible components will be located beneath the viewer under the tag 'Non-visible'.
3. We have placed a label to fill in the name of the application.
4. All buttons need to be put within the Horizontal Arrangement to keep them aligned horizontally.
5. If you have dragged and placed everything, the layout will look something like what's shown in Figure 1.

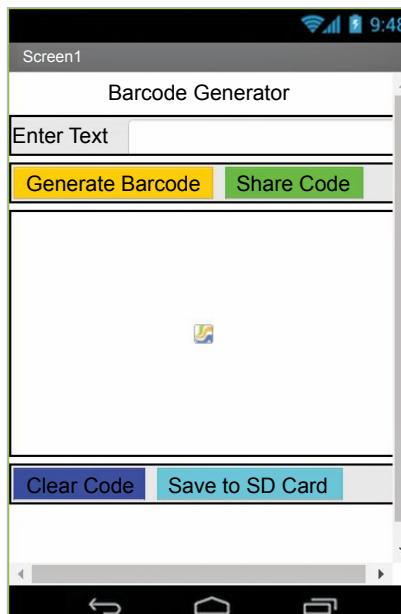


Figure 2: How the application looks

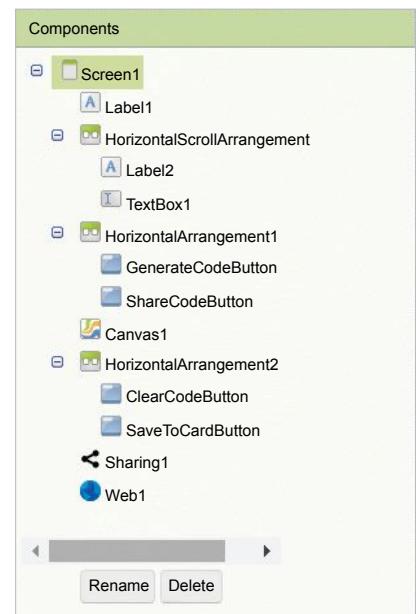


Figure 3: Components view

6. Make the necessary property changes, like we did when changing the text property for the label and button components.
  7. Renaming the components helps to identify them in the block editor.
  8. So this way, your graphical user interface is ready. Figure 2 shows how the application will look after installation to the device.
  9. Figure 3 gives the hierarchy of the components that we have dragged to the designer.
- If you are confused after seeing the designer and the components viewer, let me explain further. Here is the hierarchy that we have placed for our application.
1. At the top, we have the title of our application. It's always a good practice to name your application

and show it on the screen as well. We have put a label for it and have set its text property to the name of the application.

2. Below that we have a name label and textbox for users to type in their text. We will use this text to generate the barcode.
3. Next, we have two buttons aligned horizontally. One button will be used to trigger the event of generating code, while the other will trigger the sharing component to share it with others.
4. Next, we will display our generated barcode over the canvas area.
5. Next, there is a button to clear the canvas.
6. A Save button saves the generated image to the SD card.
7. There are sharing and Web

components for respective functions.

Now, let's head towards the blocks editor to define the various kinds of behaviour. We need to discuss the actual functionality that we are expecting from our application.

1. First, the user will be writing a text into the text box.
  2. On clicking the ‘generate code’ button, it should call the Web API to return the generated code.
  3. On clicking the ‘Share code’ button, a list should pop up with all the sharing applications being available on the device like Bluetooth, WhatsApp, Shareit, Xender, etc.
  4. The *Clear* code button should clear the canvas image.
  5. The *Save to SD card* button should

So let's move on and add these behaviours, using the block editor. I hope you remember how to switch from the designer to the block editor. There is a button available right above the *Properties* pane to help you do this.

*Block editor blocks:* I have already prepared the blocks for you. All you need to do is drag the relevant blocks from the left side palette and drop them on the viewer. Arrange the blocks in the same way as shown in Figure 4. I will explain each one, telling you what it does and how it is called.

On clicking the ‘Generate code’ button, we are calling the Web API located at <https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=>. By using the REST procedure of the Web component, we will receive the image from the URL source and the same will be set as an image on the canvas.

Let's save the image to the SD card with a common name. If the file name already exists in the SD card, it will overwrite the previous file to create a new one. Using the sharing component, we will pick the file stored in the SD card.



Figure 4: Block editor image 1

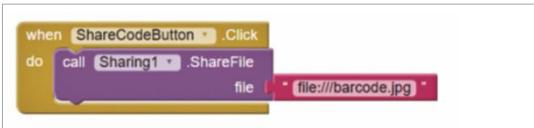


Figure 5: Block editor image 2



Figure 6: Block editor image 3

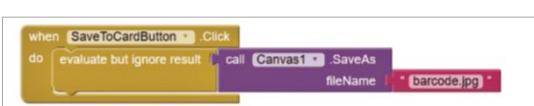


Figure 7: Block editor image 4

On clicking the ‘Clear code’ button, the canvas image should be cleared.

On selecting ‘Save to SD card’, the image should be saved to the SD card with the name *barcode.jpg*.

*Packaging and testing:* To test the app, you need to get it on your phone. First, you have to download the application to your computer and then move it to your phone via Bluetooth or USB cable. I'll tell you how to download it.

1. On the top row, click on the *Build* button. It will give you an option to download the APK to your computer.
  2. You will be able to see the progress of the download, and after it has been successfully completed, the application will be placed in the download folder of your directory or the location you have set for it.
  3. Now you need to get this APK file to your mobile phone either via Bluetooth or via USB cable. Once you have placed the APK file in your

SD card, you need to install it. Follow the on-screen instructions to install it. You might get some notification or warning saying, '*Install from un-trusted source*'. Allow this from the settings and after successful installation, you will see the icon of your application in the menu of your mobile. Here, you will see the default icon, which can be changed and we will tell you how to do that as we move ahead in this course.

I hope your application is working exactly as per your requirements. Now, depending upon your usability and customisation, you can change various things like the image, sound and behaviour, as well.

### *Debugging the application:*

We have just created the prototype of the application with very basic functionality but what else might the user be interested in? Now come various use cases, which require serious attention so as not to annoy the user, and your app should be able to address these. Consider the following cases:

1. Can we think of a way to store files with unique names?
  2. If we are saving multiple codes to the SD card, can we ask the user to browse a particular image and then share it?
  3. Can we extend the same application to read the barcode as well?

These are some scenarios that might occur and users will be pretty happy seeing these implemented.

Think about these scenarios, and how you can integrate them into the application. Do ask me if you fail to address any of the above cases.

You have successfully built another useful Android app for yourself.

Happy inventing! **END** 

By: Meghraj Singh Beniwal

The author has a B. Tech in electronics and communication, is a freelance writer and an Android app developer. He is currently working as an automation engineer at Infosys, Pune. He can be contacted at [meghrajsingh01@rediffmail.com](mailto:meghrajsingh01@rediffmail.com) or [meghrajwithandroid@gmail.com](mailto:meghrajwithandroid@gmail.com).

# An Introduction to Weka

Named after a flightless New Zealand bird, Weka is a set of machine learning algorithms that can be applied to a data set directly, or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualisation.



**M**achine learning is nothing but a type of artificial intelligence which enables computers to learn the data without help of any explicit programs. Machine learning systems crawl through the data to find the patterns and, when these are found, adjust the program's actions accordingly.

Data mining analyses the data from different perspectives and summarises it into parcels of useful information. The machine learning method is similar to data mining. The difference is that data mining systems extract the data for human comprehension. Data mining uses machine language to find valuable information from large volumes of data.

## Weka

Weka is data mining software that uses a collection of machine learning algorithms. These algorithms can be applied directly to the data or called from the Java code.

Weka is a collection of tools for:

- Regression
- Clustering

- Association
- Data pre-processing
- Classification
- Visualisation

The features of Weka are shown in Figure 1.

## Installation of Weka

You can download Weka from the official website <http://www.cs.waikato.ac.nz/ml/weka/>.

Execute the following commands at the command prompt to set the Weka environment variable for Java, as follows:

```
setenv WEKAHOME /usr/local/weka/weka-3-0-2
setenv CLASSPATH $WEKAHOME/weka.jar:$CLASSPATH
```

Once the download is completed, run the *exe* file and choose the default set-up.

## Weka application interfaces

There are totally five application interfaces available

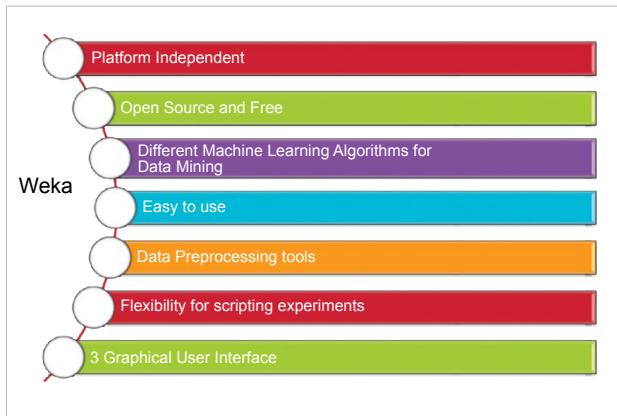


Figure 1: Weka's features

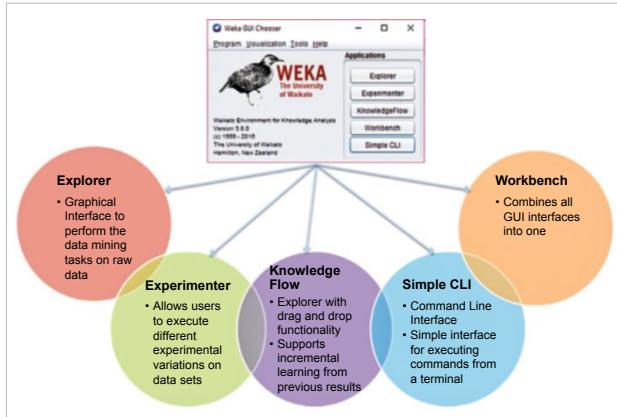


Figure 2: Weka's application interfaces

for Weka. When we open Weka, it will start the *Weka GUI Chooser* screen from where we can open the Weka application interface.

The Weka GUI screen and the available application interfaces are seen in Figure 2.

## Weka data formats

Weka uses the Attribute Relation File Format for data analysis, by default. But listed below are some formats that Weka supports, from where data can be imported:

- CSV
- ARFF
- Database using ODBC

**Attribute Relation File Format (ARFF):** This has two parts:

- 1) The header section defines the relation (data set) name, attribute name and the type.
- 2) The data section lists the data instances.

An ARFF file requires the declaration of the relation, attribute and data. Figure 3 is an example of an ARFF file.

▪ *@relation*: This is the first line in any ARFF file, written in the header section, followed by the relation/data set name. The relation name must be a string and if it contains spaces,

```
% Title: Database for fitting contact lenses
@relation lenses
@attribute age {young, pre-presbyopic, presbyopic}
@attribute spectprescrip {myope, hypermetropic}
@attribute astigmatism {no, yes}
@attribute tearprodrate {reduced, normal}
@attribute lenses {soft, hard, none}
@data
young,myope,no,reduced,none
young,myope,no,normal,soft
pre-presbyopic,myope,no,reduced,none
pre-presbyopic,myope,no,normal,soft
pre-presbyopic,myope,yes,reduced,none
presbyopic,myope,no,normal,none
presbyopic,myope,yes,reduced,none
presbyopic,myope,yes,normal,hard
```

Figure 3: An example of an ARFF file

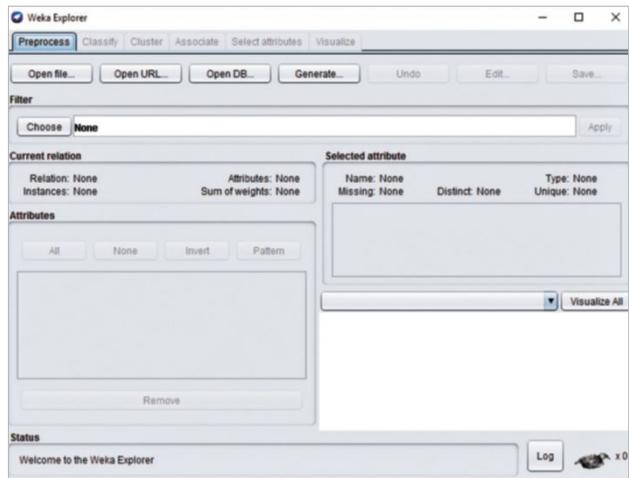


Figure 4: Weka Explorer

then it should be enclosed between quotes.

▪ *@attribute*: These are declared with their names and the type or range in the header section. Weka supports the following data types for attributes:

- Numeric
- <nominal-specification>
- String
- date
- *@data* – Defined in the *Data* section followed by the list of all data segments

## Weka Explorer

The Weka Explorer is illustrated in Figure 4 and contains a total of six tabs.

The tabs are as follows.

- 1) *Preprocess*: This allows us to choose the data file.
- 2) *Classify*: This allows us to apply and experiment with different algorithms on preprocessed data files.
- 3) *Cluster*: This allows us to apply different clustering tools, which identify clusters within the data file.
- 4) *Association*: This allows us to apply association rules, which identify the association within the data.
- 5) *Select attributes*: These allow us to see the changes on the

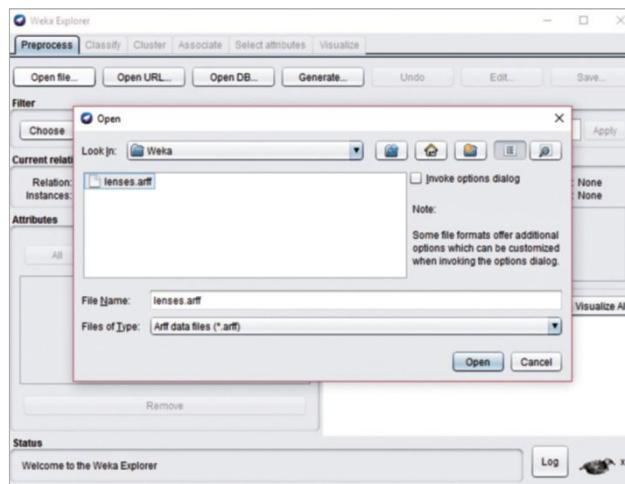


Figure 5: Preprocessing – Open data set

- inclusion and exclusion of attributes from the experiment.
- 6) **Visualize:** This allows us to see the possible visualisation produced on the data set in a 2D format, in scatter plot and bar graph output.

The user cannot move between the different tabs until the initial preprocessing of the data set has been completed.

**Preprocessing:** Data preprocessing is a must. There are three ways to inject the data for preprocessing:

- Open File – enables the user to select the file from the local machine
- Open URL – enables the user to select the data file from different locations
- Open Database – enables users to retrieve a data file from a database source

A screen for selecting a file from the local machine to be preprocessed is shown in Figure 5.

After loading the data in Explorer, we can refine the data by selecting different options. We can also select or remove the attributes as per our need and even apply filters on data to refine the result.

**Classification:** To predict nominal or numeric quantities, we have classifiers in Weka. Available learning schemes are decision-trees and lists, support vector machines, instance-based classifiers, logistic regression and Bayes' nets. Once the data has been loaded, all the tabs are enabled. Based on the requirements and by trial and error, we can find out the most suitable algorithm to produce an easily understandable representation of data.

Before running any classification algorithm, we need to set test options. Available test options are listed below.

**Use training set:** Evaluation is based on how well it can predict the class of the instances it was trained on.

**Supplied training set:** Evaluation is based on how well it can predict the class of a set of instances loaded from a file.

**Cross-validation:** Evaluation is based on cross-validation by using the number of folds entered in the 'Folds' text field.

**Split percentage:** Evaluation is based on how well it can

predict a certain percentage of the data, held out for testing by using the values entered in the '%' field.

To classify the data set based on the characteristics of attributes, Weka uses classifiers.

**Clustering:** The cluster tab enables the user to identify similarities or groups of occurrences within the data set. Clustering can provide data for the user to analyse. The training set, percentage split, supplied test set and classes are used for clustering, for which the user can ignore some attributes from the data set, based on the requirements. Available clustering schemes in Weka are k-Means, EM, Cobweb, X-means and FarthestFirst.

**Association:** The only available scheme for association in Weka is the Apriori algorithm. It identifies statistical dependencies between clusters of attributes, and only works with discrete data. The Apriori algorithm computes all the rules having minimum support and exceeding a given confidence level.

**Attribute selection:** Attribute selection crawls through all possible combinations of attributes in the data to decide which of these will best fit the desired calculation—which subset of attributes works best for prediction. The attribute selection method contains two parts.

- **Search method:** Best-first, forward selection, random, exhaustive, genetic algorithm, ranking algorithm
- **Evaluation method:** Correlation-based, wrapper, information gain, chi-squared

All the available attributes are used in the evaluation of the data set by default. But it enables users to exclude some of them if they want to.

**Visualisation:** The user can see the final piece of the puzzle, derived throughout the process. It allows users to visualise a 2D representation of data, and is used to determine the difficulty of the learning problem. We can visualise single attributes (1D) and pairs of attributes (2D), and rotate 3D visualisations in Weka. It has the Jitter option to deal with nominal attributes and to detect 'hidden' data points. 

## References

- [1] Weka Machine Learning Project, <http://www.cs.waikato.ac.nz/~ml/index.html>
- [2] E. Frank, Machine Learning With Weka, University of Waikato, New Zealand
- [3] B. Mobasher, Data Preparation and Mining with Weka, [http://maya.cs.depaul.edu/~classes/ect584/WEKA\\_association\\_rules.html](http://maya.cs.depaul.edu/~classes/ect584/WEKA_association_rules.html), DePaul University, 2003

## By: Palak Shah

The author is a senior software engineer. She loves to explore new technologies, learn innovative concepts, and is also fond of philosophy. She can be reached at [palak311@gmail.com](mailto:palak311@gmail.com).

# Analysing Sentiments with NLTK

In this article, we explore ways to analyse sentiments from a given text and how some machine learning techniques can help in the process.

**S**entiment analysis is used in opinion mining, business analytics and reputation monitoring. It helps businesses understand the customers' experience with a particular service or product by analysing their emotional tone from the product reviews they post, the online recommendations they make, their survey responses and other forms of social media text. Businesses can get feedback on how happy or dissatisfied the customer is, and use this insight to gain a competitive edge.

In this article, we explore how to conduct sentiment analysis on a piece of text using some machine learning techniques. Python happens to be one of the best programming language choices when it comes to machine learning and textual analytics as it is easy to learn, is open source, and is effective in catering to machine learning requirements like processing large data sets and performing mathematical computations. Natural Language ToolKit (NLTK) is one of the popular packages in Python that can aid in sentiment analysis.

## About NLTK

NLTK is an open source natural language processing (NLP) platform available for Python. It is capable of textual tokenisation, parsing, classification, stemming, tagging, semantic reasoning and other computational linguistics.

NLTK is a community driven project and is available for use on Linux, Mac OS X and Windows.

Let's first get started by installing NLTK to glue with Python using the following steps.

1. NLTK can be installed using Pip, a package management tool that Python users might be familiar with. Pip comes, by default, on Python version 2.7.9 and later. However, if you are using an older version of Python and don't have Pip already installed, use the following command to do so.

On Ubuntu:

```
sudo apt-get install python-pip
```



On Fedora Linux:

```
sudo yum install python-pip
```

2. With Pip, install NLTK using the following command:

```
sudo pip install -U nltk
```

This completes the NLTK download and installation, and you are all set to import and use it in your Python programs.

In this article, we will analyse sentiments from a piece of text using the NLTK sentiment analyser and the Naïve's Bayes Classifier. As a sample, I've taken some user reviews on restaurants in Bengaluru from [www.zomato.com](http://www.zomato.com) as shown below.

*"Great place to be when you are in Bangalore."*

*"The place was being renovated when I visited so the seating was limited."*

*"Loved the ambience, loved the food"*

*"The food is delicious but not over the top."*

*"Service - Little slow, probably because too many people."*

*"The place is not easy to locate"*

*"Mushroom fried rice was tasty"*

## Analysis using NLTK Vader SentimentAnalyser

NLTK comes with an inbuilt sentiment analyser module – *nltk.sentiment.vader*—that can analyse a piece of text and classify the sentences under positive, negative and neutral polarity of sentiments. A code snippet of how this could be done is shown below:

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

```
hotel_rev = ["Great place to be when you are in Bangalore.",
"The place was being renovated when I visited so the seating
```

```
was limited.",
"Loved the ambience, loved the food",
"The food is delicious but not over the top.",
"Service - Little slow, probably because too many people.",
"The place is not easy to locate",
"Mushroom fried rice was tasty"]
```

```
sid = SentimentIntensityAnalyzer()
for sentence in hotel_rev:
 print(sentence)
 ss = sid.polarity_scores(sentence)
 for k in ss:
 print('{0}: {1}, '.format(k, ss[k]), end='')
 print()
```

When running the above Python script, the different sentiment proportions for individual sentences are obtained as shown below:

Great place to be when you are in Bangalore.  
neg: 0.0, neu: 0.661, compound: 0.6249, pos: 0.339,

The place was being renovated when I visited so the seating was limited.  
neg: 0.147, neu: 0.853, compound: -0.2263, pos: 0.0,

Loved the ambience, loved the food  
neg: 0.0, neu: 0.339, compound: 0.8316, pos: 0.661,

The food is delicious but not over the top.  
neg: 0.168, neu: 0.623, compound: 0.1184, pos: 0.209,

Service - Little slow, probably because too many people.  
neg: 0.0, neu: 1.0, compound: 0.0, pos: 0.0,

The place is not easy to locate  
neg: 0.286, neu: 0.714, compound: -0.3412, pos: 0.0,

Mushroom fried rice was tasty  
neg: 0.0, neu: 1.0, compound: 0.0, pos: 0.0,

The compound value here conveys the overall positive or negative user experience.

## Analysis using Naïve's Bayes Classifier

Apart from Vader, one can create one's own classification model using Naïve's Bayes Classifier. In the machine learning context, Naïve's Bayes Classifier is a probabilistic classifier based on Bayes' theorem that constructs a classification model out of training data. This classifier learns to classify the reviews to positive or negative using the supervised learning mechanism. The learning process starts by feeding in sample data that aids the classifier to construct a model to classify these reviews.

```
import nltk
from nltk.tokenize import word_tokenize

Step 1 - Training data
train = [("Great place to be when you are in Bangalore.", "pos"),
 ("The place was being renovated when I visited so the seating was limited.", "neg"),
 ("Loved the ambience, loved the food", "pos"),
 ("The food is delicious but not over the top.", "neg"),
 ("Service - Little slow, probably because too many people.", "neg"),
 ("The place is not easy to locate", "neg"),
 ("Mushroom fried rice was spicy", "pos"),
]

Step 2
dictionary = set(word.lower() for passage in train for word in word_tokenize(passage[0]))

Step 3
t = [{(word: (word in word_tokenize(x[0]))) for word in dictionary}, x[1]] for x in train]

Step 4 - the classifier is trained with sample data
classifier = nltk.NaiveBayesClassifier.train(t)

test_data = "Manchurian was hot and spicy"
test_data_features = {word.lower(): (word in word_tokenize(test_data.lower())) for word in dictionary}

print (classifier.classify(test_data_features))
```

The output for the above code can be 'pos', denoting positive. The training data here is an array of sentences with corresponding class types – positive (pos) or negative (neg) to train the classifier. The dictionary formed in Step 2 consists of all the words obtained by tokenising or breaking this list of sentences. Step 3 starts constructing the data to be fed to the Naïve Bayes Classifier and Step 4 feeds the data to the classifier. With these steps, you might try out testing the classifier with different sentences. 

## References

- [1] <http://www.nltk.org/howto/sentiment.html>
- [2] <http://www.nltk.org/api/nltk.sentiment.html>
- [3] Hutto, C.J. and Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

## By: Shravan I.V.

The author currently works as a software developer at Cisco Systems India. He can be contacted at [iv.shravan@gmail.com](mailto:iv.shravan@gmail.com)

# An Introduction to Google Machine Learning APIs

This article gives readers an overview of Google machine learning APIs. It contains code samples to get started with APIs, bringing the power of machine learning to everyday developers.



For the past two decades, Google has been at the forefront of collecting and mining data. It has built multiple machine learning models through which it now provides some of the most powerful machine learning APIs. These machine learning models are the magic behind some of the applications like Google Photos, Inbox Reply (Gmail), Google Translate and more.

In the last year, Google has opened up its machine learning models by making its APIs accessible to the public. The Google Cloud Machine Learning Platform provides easy-to-use REST APIs that have powerful capabilities like computer vision, NLP, translation and more.

## Google Cloud Machine Learning Platform

One of the main issues facing developers is figuring out how much of theoretical knowledge they need before using machine learning algorithms in their applications. The team at Google has understood this and identified a spectrum across which different products from Google can be used from the Machine Learning Platform.

At one end of the spectrum is TensorFlow, which is an open source library for machine intelligence. TensorFlow does require that you are up to speed on various machine learning concepts and can use the API to build machine models, using them effectively in your applications. This might not be possible for all developers, especially those who are looking for REST APIs to use in their applications. To cater to this need, Google has identified multiple capabilities

in its Machine Learning Platform and has exposed those capabilities in the form of easy-to-call REST APIs. There is also a free quota per month for these APIs.

Over the last few months, Google has released multiple REST APIs in the Machine Learning Platform. The most important ones are listed below.

**Google Cloud Vision API:** The Cloud Vision API provides a REST API to understand and extract information from an image. It uses powerful machine learning models to classify images into thousands of categories, detect faces, identify adult content, emotions, OCR support and more. If you are looking to classify and search through images for your application, this is a good library to consider.

**Google Natural Language API:** The Natural Language API is used to identify parts of speech and to detect multiple types of entities like persons, monuments, etc. It can also perform sentiment analysis. It currently supports three languages: English, Spanish and Japanese.

**Google Speech API:** The Speech API is used to translate audio files into text. It is able to identify over 80 languages and their variants, and can work with most audio files.

There are multiple other APIs in this family like the Translate API, but we will limit our sample code to the above three APIs.

## Getting started

To get started with using the machine learning APIs listed above, you need to sign up for the Google Cloud Platform and

enable your billing. The platform provides a trial period of two months for US\$300, which is sufficient to try out the APIs.

Once you have signed up for the platform, you need to create a project. Go to the *Cloud Console* and click on *New Project*. In the *New Project* dialogue box, provide a name for your project, as shown in Figure 1, before clicking on *Create*.

Once the project is created, you need to enable the above three APIs for your project. To do that, go to the *Main Menu* on the top left in the *Cloud Console* and click on *API Manager*, as shown in Figure 2.

In the search bar, enter 'Cloud Vision' and it will display the Google Cloud Vision API in the list. Select it and then click on *Enable*. This will enable the API. Perform the same operation for the other two APIs also, i.e., the natural language processing (NLP) API and the Speech API. Ensure that you have enabled the above three APIs for your project before moving ahead.

Since we are going to be writing an external program that invokes the REST APIs, our calling client application needs to identify itself to the Google Cloud platform. We do that through the concept of a service account, which will be used by our client application to invoke the APIs.

To do that, we need to download the service account keys for our project. From the *API Manager* menu on the left, click on *Credentials* and then on *Create Credentials*. Select *Service Account Key* from the options. This will bring up a dialogue box as shown in Figure 3. Select *New Service*

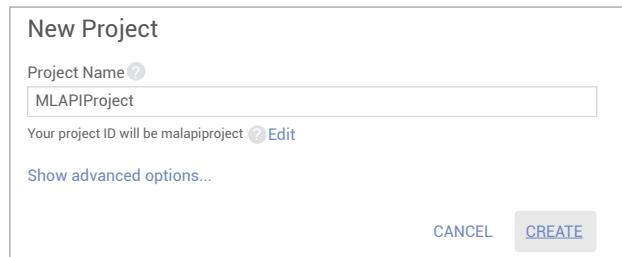


Figure 1: *New Project* dialogue box

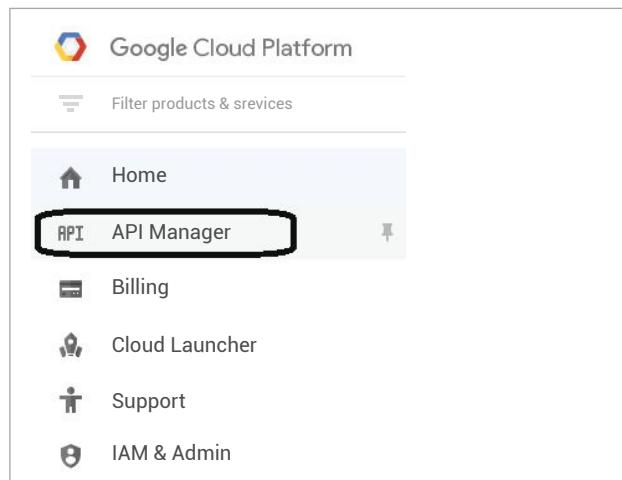


Figure 2: *API Manager*

*Account*, give a name to the service account and in *Role*, select *Project Owner*.

Click on *Create*. This will download a JSON file for your service account. Ensure that you save this file in a safe place, since we will use it when we run our client application. For the purpose of this article, the service account JSON file downloaded will be referred to as the *mlapi-serviceaccount.json* file.

## Vision API

Let's first test the vision API. As mentioned earlier, this has multiple features and, in this specific example, we are going to test its label detection feature. This feature allows you to provide an image, and it will identify various labels for this image, which can help in its classification.

We are going to provide it an image of a cricket match and see what it can figure out about it. The image that we are going to test it against is shown in Figure 4.

Before you run this code and the other examples, you will need to set an environment variable named *GOOGLE\_APPLICATION\_CREDENTIALS* and it will need to point to the service account

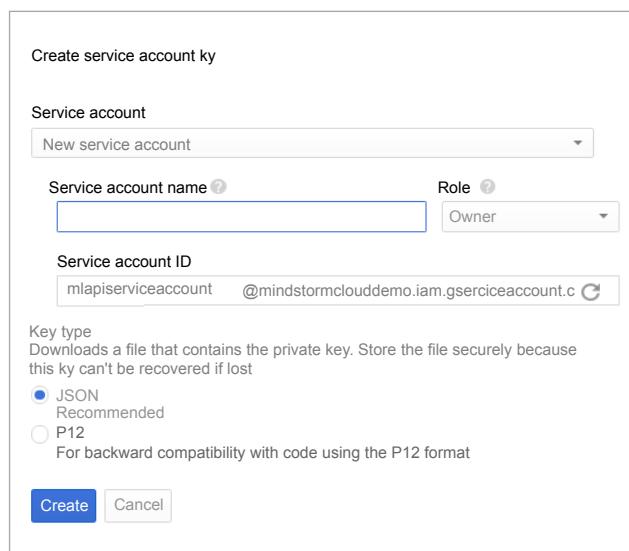


Figure 3: *Service Account Key* creation

JSON file that we downloaded earlier. For example:

```
export GOOGLE_APPLICATION_CREDENTIALS=mlapi-serviceaccount.json
```

The sample code in Python is shown below. This is a modification of the code that is available at its API page.

```
from googleapiclient import discovery
import argparse
import base64
import json
from oauth2client.client import GoogleCredentials
```



Figure 4: Cricket image

```

DISCOVERY_URL = ('https://{}{api}.googleapis.com/'
 '$discovery/rest?version={apiVersion}')

def main(photo_file):
 credentials = GoogleCredentials.get_application_default()
 service = discovery.build('vision',
 'v1', credentials=credentials)
 with open(photo_file, 'rb') as image:
 image_content = base64.b64encode(image.read())
 service_request = service.images().annotate(body={
 'requests': [
 {
 'image': {
 'content': image_content.decode('UTF-8')
 },
 'features': [
 {
 'type': 'LABEL_DETECTION',
 'maxResults': 5
 }
]
 }
]
 })
 response = service_request.execute()
 print(json.dumps(response['responses'], indent=4))

if __name__ == '__main__':
 parser = argparse.ArgumentParser()
 parser.add_argument('image_file', help='The image you\'d
like to label.')
 args = parser.parse_args()
 main(args.image_file)

```

Let us understand the code first. In the main block, we read the file name that we provided. The first step in the main function is to establish the credentials; we use the standard application credentials, which were set via the environment variable. We then read the file bytes and invoke the REST API endpoint for the cloud vision API. Notice that in the REST API call, we provide not just the bytes for the image but also the feature that we are interested in, i.e., *LABEL\_DETECTION*, and command it to return us a maximum of five such label results. We finally invoke the API and print out the results.

The output can be downloaded from [http://opensourceforu.com/article\\_source\\_code/dec16/gml.zip](http://opensourceforu.com/article_source_code/dec16/gml.zip). Notice that each of the labels that it detected has a confidence score out of 10. It is generally considered a positive response if it is about 70-80 per cent, but it is entirely up to your application to accept these results. Note that it has also

detected erroneously that there is a baseball player—it is important to keep these points in mind.

## Natural Language API

The Natural Language API helps us to identify parts of speech in a text, entities (persons, monuments, organisations, etc) and also do sentiment analysis. In the specific example that we will execute here, we are going to test how the API extracts entities. This will help us to identify the different entities that were present in the piece of text that we provided to it.

The text that we will provide to the REST API is as follows:

*The search for a new chairman of Tata Sons following the ouster of Cyrus Mistry has thrown up quite a few names. The high-profile names include Indra Nooyi, the head of PepsiCo Inc, Arun Sarin, the former head of Vodafone Group, and Noel Tata, chairman of the Tata retail unit, Trent.*

The code is shown below and it follows a pattern similar to the vision API that we saw earlier.

```

from googleapiclient import discovery
import httplib2
import json
from oauth2client.client import GoogleCredentials

DISCOVERY_URL = ('https://{}{api}.googleapis.com/'
 '$discovery/rest?version={apiVersion}')

def main():

 http = httplib2.Http()

 credentials = GoogleCredentials.get_application_default().create_scoped(
 ['https://www.googleapis.com/auth/cloud-platform'])

 http=httplib2.Http()
 credentials.authorize(http)

 service = discovery.build('language', 'v1beta1',
 http=http,
 discoveryServiceUrl=DISCOVERY_URL)

 service_request = service.documents().analyzeEntities(
 body={
 'document': {
 'type': 'PLAIN_TEXT',
 'content': "The search for a new chairman of Tata
Sons following the ouster of Cyrus Mistry has thrown up quite
a few names. The high-profile names include Indra Nooyi, the
head of PepsiCo Inc, Arun Sarin, the former head of Vodafone
Group, and Noel Tata, chairman of the Tata retail unit
Trent."
 }
 }
)

```

```
})

response = service_request.execute()
entities = response['entities']
print(json.dumps(entities, indent=4))
return 0

if __name__ == '__main__':
 main()
```

Once the credentials are established, we are simply going to invoke the REST API and, specifically, the *analyzeEntities* method. The output results can be obtained from [http://open sourceforu.com/article\\_source\\_code/dec16/gml.zip](http://open sourceforu.com/article_source_code/dec16/gml.zip).

It is interesting to see that the Natural Language API has been able to correctly identify the entities (organisations and individuals) and, in some cases, also provided information on their Wikipedia entries.

## Speech API

The Speech API is used to convert a RAW audio file to its equivalent text format. The API is powerful enough to work with disturbances in the audio and has multiple optimisations inbuilt to detect the text.

The code shown below is taken from the official documentation, and the parameter that we pass to it at runtime is the audio file in RAW format. The code simply reads the audio file bytes, provides information in the request about the kind of audio and then, in a synchronous fashion, gets back the results of the audio converted to text.

```
import argparse
import base64
import json
from googleapiclient import discovery
import httplib2
from oauth2client.client import GoogleCredentials
[START authenticating]
DISCOVERY_URL = ('https://{}.googleapis.com/$discovery/rest?'
 'version={apiVersion}')
def get_speech_service():
 credentials = GoogleCredentials.get_application_default().create_scoped(
 ['https://www.googleapis.com/auth/cloud-platform'])
 http = httplib2.Http()
 credentials.authorize(http)

 return discovery.build(
 'speech', 'v1beta1', http=http,
 discoveryServiceUrl=DISCOVERY_URL)
def main(speech_file):
 with open(speech_file, 'rb') as speech:
 speech_content = base64.b64encode(speech.read())
```

```
service = get_speech_service()
service_request = service.speech().syncrecognize(
 body={
 'config': {
 'encoding': 'LINEAR16',
 'sampleRate': 16000,
 'languageCode': 'en-US',
 },
 'audio': {
 'content': speech_content.decode('UTF-8')
 }
 })
response = service_request.execute()
print(json.dumps(response))
if __name__ == '__main__':
 parser = argparse.ArgumentParser()
 parser.add_argument(
 'speech_file', help='Full path of audio file to be
 recognized')
 args = parser.parse_args()
 main(args.speech_file)
```

The output is shown below, and is the actual audio content that was present in the file.

```
{"results": [{"alternatives": [{"confidence": 0.98267895,
"transcript": "how old is the Brooklyn Bridge"}]}]}
```

Google Machine Learning APIs provide a wide range of APIs that developers can use today to incorporate powerful machine learning algorithms in the domains of computer vision, language analysis, sentiment analysis, audio conversion and translation. The key factors that make this a compelling list of APIs is Google's track record in machine learning—the complex machine learning models that it has built over the last decade—and the ease and accuracy of the APIs. 

## References

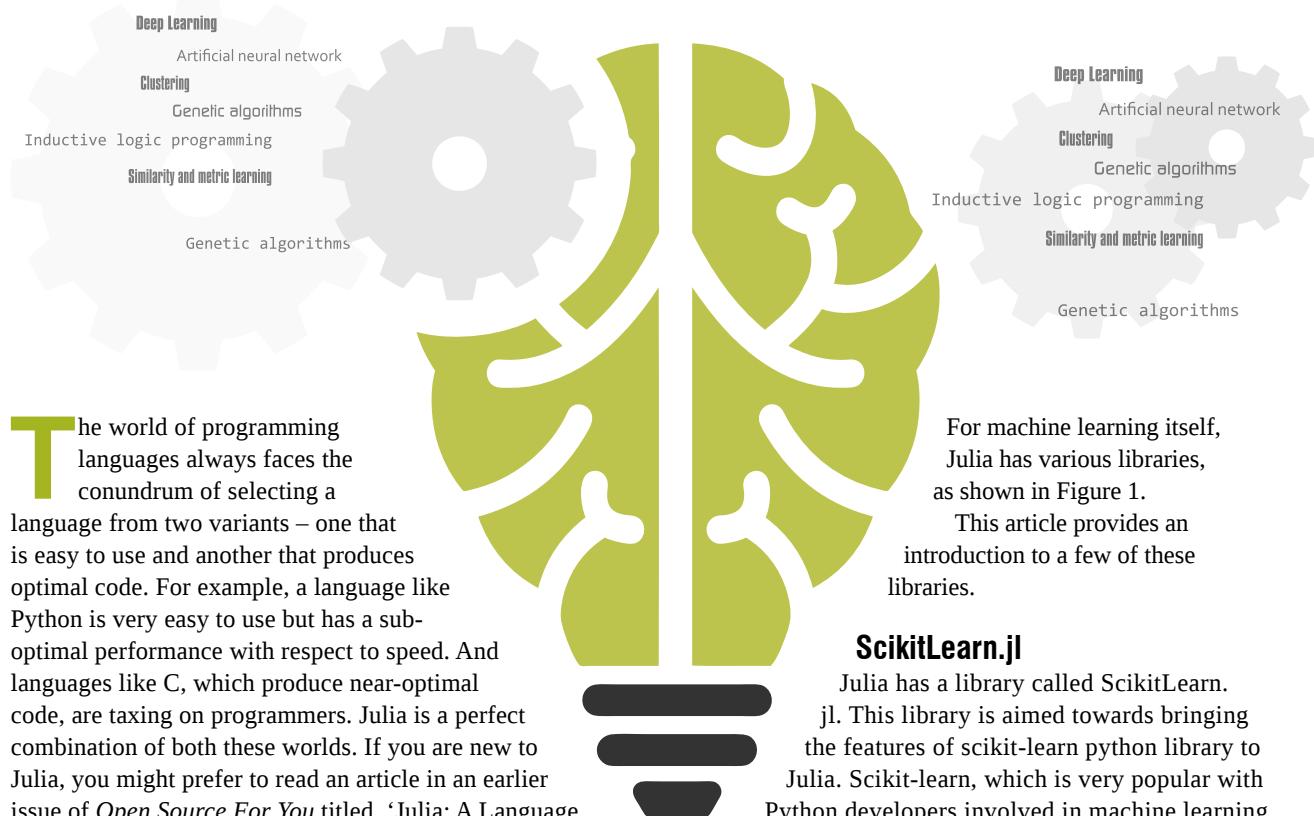
- [1] Google Cloud Machine Learning: <https://cloud.google.com/products/machine-learning/>
- [2] Google Cloud Vision API: <https://cloud.google.com/vision/>
- [3] Google Natural Language API: <https://cloud.google.com/natural-language/>
- [4] Google Speech Recognition API: <https://cloud.google.com/speech/>
- [5] TensorFlow: <https://www.tensorflow.org/>
- [6] Google Cloud Free Trial: <https://console.cloud.google.com/freetrial>

## By: Romin Irani

The author has been working in the software industry for 20+ years. His passion is to read, write and teach about technology in order to make developers successful. He blogs at [www.rominirani.com](http://www.rominirani.com).

# Machine Learning: The Julia Approach

Julia is a rising star in the world of programming languages, having inherited the best features of both Python and C. This article provides insights into the machine learning related features offered by Julia. As it is optimised for both speed and ease-of-use, developing machine learning based applications in Julia is fairly straightforward.



The world of programming languages always faces the conundrum of selecting a language from two variants – one that is easy to use and another that produces optimal code. For example, a language like Python is very easy to use but has a sub-optimal performance with respect to speed. And languages like C, which produce near-optimal code, are taxing on programmers. Julia is a perfect combination of both these worlds. If you are new to Julia, you might prefer to read an article in an earlier issue of *Open Source For You* titled, ‘Julia: A Language that Walks like Python and Runs like C’. In addition to this, there are plenty of resources on the Web that enable a developer to become familiar with Julia (<http://docs.julialang.org/en/release-0.5/>).

The focus of this article is to introduce readers to machine learning and associated tasks in Julia. The core idea of machine learning is to enable machines to model the problem space in such a way that programs are capable of handling novel scenarios. For example, an artificial neural network for digit recognition should be able to recognise the variations of digits that were not present in the training dataset. In simple words, these networks are bio-inspired, similar to the manner in which biological creatures respond to scenarios, after successful training. Machine learning can be used across various domains such as text recognition, image recognition, classification, etc.

Similar to Python, Julia is also extensible. It has many libraries and packages focusing on specific domains.

For machine learning itself, Julia has various libraries, as shown in Figure 1.

This article provides an introduction to a few of these libraries.

## ScikitLearn.jl

Julia has a library called *ScikitLearn.jl*. This library is aimed towards bringing the features of *scikit-learn* python library to Julia. *Scikit-learn*, which is very popular with Python developers involved in machine learning based projects and has implemented the *scikit-learn* algorithms. The major features of *ScikitLearn.jl* are listed below:

- It has around 150 models accessed with a uniform interface
- Pipelines and FeatureUnions
- Cross-validation
- Hyperparameter tuning
- DataFrames support

Installation of *ScikitLearn.jl* may be done using the following command:

```
Pkg.add("ScikitLearn")
```

It has to be noted that the *scikit-learn* Python library is required to import the optional Python modules. *Pyplot.jl* is also utilised. As noted in the official documentation, *ScikitLearn.jl* is not associated with *scikit-learn.org*.

A simple classifier example with the Iris dataset and LogisticRegression is shown below (<http://scikitlearnjl.readthedocs.io/en/latest/quickstart/>):

```
The Rdataset shall be added, if unavailable.
using RDatasets: dataset

iris = dataset("datasets", "iris")
ScikitLearn.jl expects arrays, but DataFrames can also be
used - see
the corresponding section of the manual
X = convert(Array, iris[[:SepalLength, :SepalWidth,
:PetalLength, :PetalWidth]])
y = convert(Array, iris[:Species])

Load the Logistic Regression model
using ScikitLearn
This model requires scikit-learn. See
http://scikitlearnjl.readthedocs.io/en/latest/models/#installation
@sk_import linear_model: LogisticRegression
The Hyperparameters such as regression strength, whether to
fit the intercept, penalty type.
model = LogisticRegression(fit_intercept=true)

Train the model.
fit!(model, X, y)

Accuracy is evaluated
accuracy = sum(predict(model, X) .== y) / length(y)
println("accuracy: $accuracy")
```

The cross-validation is shown in the code given below:

```
using ScikitLearn.CrossValidation: cross_val_score

cross_val_score(LogisticRegression(), X, y; cv=5) # 5-fold
> 5-element Array{Float64,1}:
> 1.0
> 0.966667
> 0.933333
> 0.9
> 1.0
```

*ScikitLearn.jl* may be used with other classifiers such as the DecisionTree classifier (*DecisionTree.jl* - <https://github.com/bensadeghi/DecisionTree.jl>). A random Forest based classifier is also shown below:

```
using RDatasets: dataset
using DecisionTree
iris = dataset("datasets", "iris")
features = convert(Array, iris[:, 1:4]);
labels = convert(Array, iris[:, 5]);
```

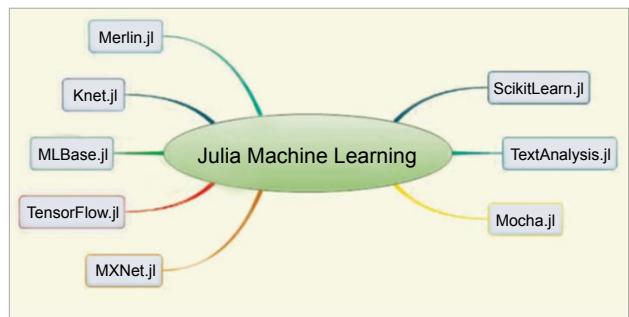


Figure 1: Machine learning libraries in Julia

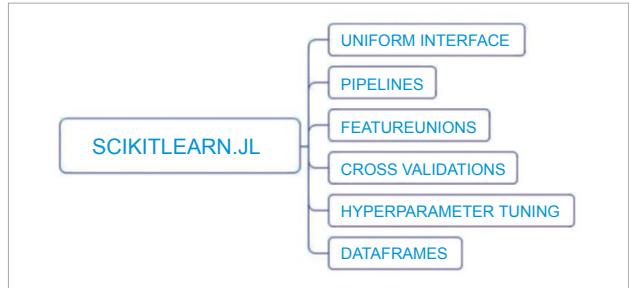


Figure 2: Features of *ScikitLearn.jl*

```
train full-tree classifier
model = DecisionTreeClassifier(pruning_purity_threshold=0.9,
maxdepth=6)
fit!(model, features, labels)
pretty print of the tree, to a depth of 5 nodes (optional)
print_tree(model.root, 5)
apply learned model
predict(model, [5.9,3.0,5.1,1.9])
get the probability of each label
predict_proba(model, [5.9,3.0,5.1,1.9])
println(get_classes(model)) # returns the ordering of the
columns in predict_proba's output
run n-fold cross validation over 3 CV folds
See ScikitLearn.jl for installation instructions
using ScikitLearn.CrossValidation: cross_val_score
accuracy = cross_val_score(model, features, labels, cv=3)
```

The following models are available in *DecisionTree.jl*:

- DecisionTreeClassifier
- DecisionTreeRegressor
- RandomForestClassifier
- RandomForestRegressor
- AdaBoostStumpClassifier

## TensorFlow

TensorFlow is a recent buzzword in the machine learning world. This active open source machine learning framework from Google focuses on numerical computation with data flow graphs. The nodes represent mathematical operations, and the edges connecting the node indicate the data arrays termed *Tensors*. *TensorFlow.jl* is a Julia

wrapper for TensorFlow, which can be installed using *Pkg.add*, as shown below:

```
Pkg.add("TensorFlow")
```

Using the GPU for efficient processing is an important attribute of machine learning approaches. To activate the GPU support, the environment variable *TF\_USE\_GPU* needs to be set to 1. To enable GPU usage, CUDA 7.5 and *cudnn* are needed.

```
ENV["TF_USE_GPU"] = "1"
Pkg.build("TensorFlow")
```

It has to be noted that the TensorFlow API is huge, and all the functionalities are not wrapped at present. Some of the currently available functionalities are listed below:

- Unary and binary mathematical functions
  - Commonly used neural network operations. This covers convolutions, recurrent neural networks, etc
  - Fundamental image-loading and resizing operations
- The following are not currently wrapped:
- Control flow operations
  - Distributed graph execution
  - PyBoard graph visualisation

A code fragment to classify MNIST digits with TensorFlow is provided in this section (<https://malmaud.github.io/tfdocs/tutorial/>).

1. The TensorFlow session is built as shown below:

```
using TensorFlow
sess = Session()
```

2. A Softmax Regression Model is built as follows:

```
x = placeholder(Float32)
y = placeholder(Float32)
W = Variable(zeros([784, 10]))
b = Variable(zeros([10]))
run(sess, initialize_all_variables())
y = nn.softmax(x*W + b)
cross_entropy = reduce_mean(-reduce_sum(y_ .* log(y),
reduction_indices=[2]))
```

3. Training of the model is shown below:

```
train_step = train.minimize(train.
GradientDescentOptimizer(.00001), cross_entropy)
for i in 1:1000
 batch = next_batch(loader, 100)
 run(sess, train_step, Dict(x=>batch[1], y_=>batch[2]))
end
```

4. The model evaluation is done using the following code snippet:

```
correct_prediction = indmax(y, 2) .== indmax(y_, 2)
accuracy=reduce_mean(cast(correct_prediction, Float32))
testx, testy = load_test_set()

println(run(sess, accuracy, Dict(x=>testx, y_=>testy)))
```

## MXNet

MXNet is an active deep learning package available in Julia. It facilitates efficient GPU handling for optimal results.

*MXNet.jl* is the Julia version of *dmlc/mxnet* (<https://github.com/dmlc/mxnet>). The major features of *MXNet.jl* are:

- It enables efficient tensor computation; it can efficiently handle multiple computing devices, GPUs and distributed nodes.
- It facilitates effective manipulation of deep learning models.
- A simple three-layer MLP can be defined as follows:

```
using MXNet
data = mx.Variable(:data)

fc1 = mx.FullyConnected(data = data, name=:fc1, num_
hidden=128)
act1 = mx.Activation(data = fc1, name=:relu1, act_type=:relu)
fc2 = mx.FullyConnected(data = act1, name=:fc2, num_
hidden=64)
act2 = mx.Activation(data = fc2, name=:relu2, act_type=:relu)
fc3 = mx.FullyConnected(data = act2, name=:fc3, num_
hidden=10)
```

The aforementioned code fragment forms a feedforward chain. The recognition of digits 0 to 9 is handled with 10 output classes. A complete tutorial sequence is available at <http://dmlc.ml/MXNet.jl/latest/tutorial/mnist/#Convolutional-Neural-Networks-1>.

Machine learning incorporates a large spectrum of approaches. This article is just the tip of the iceberg. If you are interested in learning more about the topic, then the following link has pointers to plenty of resources in one place: <https://github.com/josephmisiti/awesome-machine-learning>. 

## References

- [1] <https://github.com/dmlc/mxnet>
- [2] <http://scikitlearnjl.readthedocs.io/en/latest/quickstart/>
- [3] <https://github.com/josephmisiti/awesome-machine-learning>
- [4] <https://github.com/malmaud/TensorFlow.jl>

## By: Dr K.S. Kuppusamy

The author is assistant professor of computer science at the Pondicherry Central University. He has 11+ years of experience in academia, research and industry. He can be reached via mail at [kskuppu@gmail.com](mailto:kskuppu@gmail.com).

# Ruby on Rails: A Powerful Open Source Web Framework for Beginners

Rails is a Web application framework designed to work with the Ruby programming language.



Web developers or those who like to build websites have probably heard about Ruby on Rails (RoR).

For those who aren't familiar with it, let's explore this magical and powerful Web framework. RoR is a full stack, which supports both the front-end and back-end. Some popular clients of this framework are Twitter, Basecamp and Amazon.

## What is Ruby on Rails?

In layman terms, Ruby is a language and Ruby on Rails is a framework built on the former. It is also known as just Rails.

Ruby on Rails aims to reduce the complexity of building powerful Web applications. It provides a default structure for Web application code, databases and all the Web pages. So, rather than managing all the configurations, Web developers can quickly start building on their innovative ideas.

David Heinemeier Hanson (a.k.a. as DHH) created the Ruby on Rails framework. In one of his interviews, he said that Ruby's conciseness, ease-of-use and high level design patterns attracted him to it rather than PHP.

Ruby is a highly object-oriented, user-friendly, general-

purpose, dynamic programming language developed in the mid-90s. If you don't know programming, then it's the best language to start with. Many startups prefer to use RoR for faster and cost-effective development. The ultimate goal of RoR is to keep developers happy while coding. It is so elegant and clean that anyone can easily come to grips with it. Other languages in the market like Java and .NET are not very suited for rapid development and also involve costs. Here are a few benefits of using Ruby on Rails.

- **Open source**  
Anyone can use it and modify it according to one's needs, but along with flexibility, it has its own naming convention, which makes this language elegant and clean.
- **It reduces the time to market drastically**  
Web development in RoR is one or two times faster than using traditional frameworks.
- **Highly compatible**  
Development speed increases if the framework is compatible with other platforms. With the increasing popularity and benefits of cloud deployments, startups are moving into cloud based deployment to reduce initial deployment costs.

and for faster development. Companies are using PaaS platforms like Heroku to reduce infrastructure investments. Luckily, RoR supports all the major PaaS platforms. It also incorporates RESTful architecture. RoR handles RESTful APIs properly, so developers can focus on the main logic rather than the API build-up.

- **It follows the MVC design pattern**

RoR follows the Model View Controller (MVC) design where the model can be co-related with database-side coding, the view is for Web pages/HTML templates, and the controller acts as a mediator between the front-end and the back-end.

- **Strong community support**

Ruby on Rails has very strong community support. Whenever beginners join the RoR community, they get plenty of help from everyone. Along with that, RoR has a very rich set of free plugins, which are ready-made functionalities and are ready to use. Rather than building from scratch, developers can use these plugins to speed up their development process.

- **Object-oriented**

In RoR, everything is an object. It is highly object-oriented and easy to learn. Even numbers and strings are objects.

## Installation steps

**Windows:** To install Ruby on Rails, we need to install components like Ruby, Rails, Git, SQLite and some of the other components that Rails needs to be able to run on Windows. It is a cumbersome task, but *railsinstaller* has made our lives easier. We have to go to <http://railsinstaller.org/> and just click on *Installer*. Currently, the latest version is *railsinstaller-3.2.0.exe*.

Mac users can also use this URL as a reference to install Ruby on Rails.

Figure 1 demonstrates the page from where we can download *railsinstaller*, which will automatically install all the components to run Ruby on Rails.

After installing Rails, try to run *rails -v* to check the version of Rails installed. Sometimes, you may get an error like the one shown below:

*"The system cannot find the path specified."*

This is because some *.bat* files have the wrong path.

To correct this, simply open your favourite text editor—in my case it's the Sublime text editor—and take the following steps.

Open all the files under *C:\RailsInstaller\Rb2.2.0\bin* in the text editor and use the *Find and replace* option (the shortcut key for the Sublime text editor is: *Ctrl+Shift+F*). Now, replace *@ C:\Users\emachnic\GitRepos\railsinstaller-windows\stage\Rb2.2.0\bin\ruby.exe* with *@%~dp0ruby.exe*.

Once it is done, you can run the *rails -v* command and

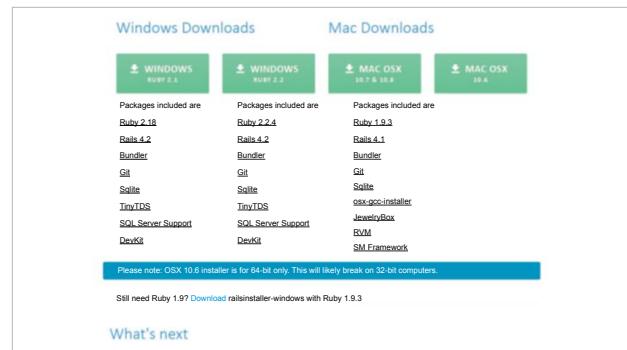


Figure 1: Ruby on Rails: Windows installation

you should not get any error message.

**Linux (Ubuntu):** Resolving software dependencies, installation and maintaining software versions successfully sometimes becomes difficult in Linux based distributions. But, RVM (Ruby Version Manager) has made our lives easy while installing Ruby on Rails.

Ubuntu is the most popular Linux distribution available in the market, so considering that, we have demonstrated the installation steps in Ubuntu OS.

To install Ruby on Rails with RVM, run the following commands as a normal user:

```
gpg --keyserver hkp://keys.gnupg.net --recv-keys
409B6B1796C275462A1703113804BB82D39DC0E3
curl -sSL https://get.rvm.io | bash -s stable --rails
```

Enter the normal user password to proceed further.

Now, let's see how this command works.

The *gpg* command is used to connect with a public key server and to request a key attached with that particular ID. Here, we are fetching the RVM project's key, which is utilised to sign every RVM release. Having the RVM project's public key permits us to check the legitimacy of the release, which we will download now and sign with the matching private key.

The remaining portion of the command, starting with *curl*, uses the Curl utility to grab a file from the RVM website. It is a script that will automatically install all related components. Here "*\*" is not an alias for any command.

The *-s* flag is for silent mode, and the *-S* flag overrides it to display errors if Curl fails. The *-L* flag is for redirection.

The script is then channelised to the bash prompt to run. The *-s* flag indicates that the input is coming from *standard in*. The rest of the commands are used to install the latest and most stable version of RVM and Rails, along with Ruby.

After this process, we have to source the RVM scripts by using the command below:

```
source ~/.rvm/scripts/rvm
```

The Ruby on Rails framework is now installed and we can start building a Web application. To install a specific version

```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\maparekh>
C:\Users\maparekh>
C:\Users\maparekh>rails -v
Rails 4.2.5.1

C:\Users\maparekh>ruby -v
ruby 2.2.4p230 (2015-12-16 revision 53155) [i386-mingw32]
C:\Users\maparekh>
```

Figure 2: Default Web page for the created application

```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\maparekh>
C:\Users\maparekh>
C:\Users\maparekh>cd Desktop
C:\Users\maparekh\Desktop>mkdir RoR_projects
The syntax of the command is incorrect.
C:\Users\maparekh\Desktop>mkdir RoR_projects
C:\Users\maparekh\Desktop>cd RoR_projects
C:\Users\maparekh\Desktop\RoR_projects>
```

Figure 3: Opening a project in the Sublime text editor

of Ruby and Rails, follow the link <https://www.digitalocean.com/community/tutorials/how-to-install-ruby-on-rails-on-ubuntu-14-04-using-rvm>.

Once the installation is completed, a Windows user can go to the command prompt and an Ubuntu user to the terminal, and run the following commands to check the version of Ruby and Rails:

```
rails -v
ruby -v
```

Here, the `-v` option is to check the version of a particular package.

## Building a Web application

Let's build a sample Web application and later we can look at how we can convert it into a powerful static Web application. To avoid information overload, I will not go too deep into this topic. We will skip the database connectivity part, which we will cover in subsequent articles on RoR.

```
Administrator: C:\WINDOWS\system32\cmd.exe
C:\Users\maparekh\Desktop>cd RoR_projects
C:\Users\maparekh\Desktop>RoR_projects>rails new myfirstapp
 create README.rdoc
 create Rakefile
 create config.ru
 create .gitignore
 create Gemfile
 create app
 create app/assets/javascripts/application.js
 create app/assets/stylesheets/application.css
 create app/controllers/application_controller.rb
 create app/helpers/application_helper.rb
 create app/views/layouts/application.html.erb
 create app/assets/images/.keep
 create app/mailers/.keep
 create app/models/.keep
 create app/controllers/concerns/.keep
 create app/models/concerns/.keep
 create bin
 create bin/bundle
 create bin/rails
 create bin/rake
 create bin/setup
 create config
 create config/routes.rb
 create config/application.rb
 create config/environment.rb
 create config/secrets.yml
 create config/environments
 create config/environments/development.rb
 create config/environments/production.rb
 create config/environments/test.rb
 create config/initializers
 create config/initializers/assets.rb
 create config/initializers/backtrace_silencers.rb
```

Figure 4: Changes in `routes.rb` to see our sample home page of the application

```
Administrator: C:\WINDOWS\system32\cmd.exe - rails server
C:\Users\maparekh\Desktop\RoR_projects\myfirstapp>rails server
 Booting WEBrick...
 => Rails 4.2.5.1 application starting in development on http://localhost:3000
 => Run 'rails server -h' for more startup options
 => Ctrl-C to shutdown server
[2016-08-12 09:32:41] INFO WEBrick 1.3.1
[2016-08-12 09:32:41] INFO Ruby 2.2.4 (2015-12-16) [i386-mingw32]
[2016-08-12 09:32:41] INFO WEBrick::HTTPServer#start: pid=12512 port=3000
```

Figure 5: Sample home page of the application

Before creating a project, first create or select a directory in which you want to store all RoR projects. In my case, I have created a directory called `RoR_projects`.

```
mkdir RoR_projects
cd RoR_projects
```

After that, we will create a new project, which will be our first Web application on RoR.

```
rails -new myfirstapp
```

Once we run this command, it will create everything that is required to build a Web application for us. The entire structure will be created. When we go into the `myfirstapp` folder and then go to `dir` (in Windows) and `/ls` (in Linux), we can see different folders and files already created inside it.

Here, my path is `C:\Users\maparekh\Desktop\RoR_projects\myfirstapp`.

If we want to check the app we created, we have to first go to our project directory and then run the local server. RoR has its own server called `WEBrick` and besides, it comes with `SQLite`, by default, which is a serverless database.

We have to make sure that the server is up and running in another command prompt/terminal to check if the application runs:

```
rails server
```

Now, copy the URL `http://localhost:3000` and paste it in a browser. We can see the default Web application page, which

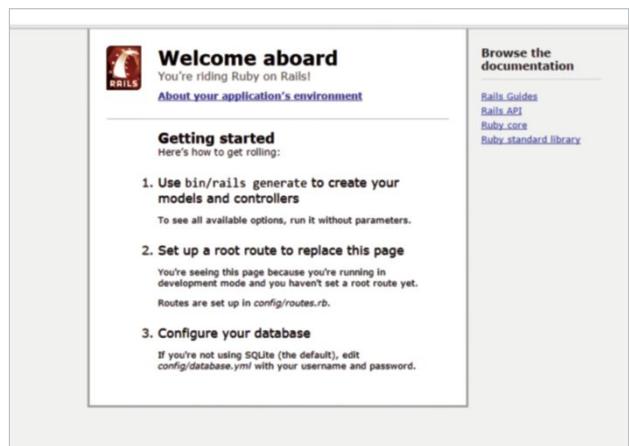


Figure 6: Modified home page of application

```

Administrator: C:\WINDOWS\system32\cmd.exe
C:\Users\maparekh\Desktop\RoR_projects>cd myfirstapp
C:\Users\maparekh\Desktop\RoR_projects\myfirstapp>rails generate controller welcome homepage
 create app/controllers/welcome_controller.rb
 route get 'welcome/homepage'
 invoke erb
 create app/views/welcome
 create app/views/welcome/homepage.html.erb
 invoke test_unit
 create test/controllers/welcome_controller_test.rb
 invoke helper
 create app/helpers/welcome_helper.rb
 invoke test_unit
 create test/assets
 create coffee
 create app/assets/javascripts/welcome.coffee
 invoke scss
 create app/assets/stylesheets/welcome.scss
C:\Users\maparekh\Desktop\RoR_projects\myfirstapp>

```

Figure 7: Copying contents from the template to our *myfirstapp* application

means that the app is up and running.

Now, instead of the default page, we use our own home page.

To do that, we have to create a controller that is used as a mediator to go to a specific Web page. Basically, it will control the calls and tell us *when* we have to call or display which Web page.

Here is the command to create the controller and Web page:

```
rails generate controller welcome homepage
```

‘Generate’ is the keyword used to create the controller, ‘welcome’ is a controller name and ‘homepage’ is the view created under this controller.

Now open your project folder. You will find *myfirstapp* in the text editor (I am using Sublime).

Given below is some information about the contents of the folder and the files associated with it.

**App:** App is the folder in which we will spend most of our time. It contains the MVC architecture, and there are folders like views, models, mailers, helpers, controllers and assets. Apart from what we already know about MVC, there are helpers, which are small functions or modules used in applications; while assets contain images, CSS and JavaScript files.

**Bin:** This folder contains all the commands and the set-up utility. Usually, we will not touch this folder.

**Config:** This folder contains all the configuration files including *database.yml*, which is for database related configuration and *routes.rb*, where we will write routes to call Web pages.

All other folders are of not much use currently, so we will not go deeper into them.

Now, to check our home page, we need to do some modifications. After running the previous controller command, we can look into the folders that are inside the controller and also the views. There are some new files created.

If you can't see the new files created, go to your folder and press F5 or refresh it. Then check for newly created files.

As you can see in Figure 4, we need to create routes so that the request will be directed to our home page. Check line no. 8 inside the *Config/routes.rb* file, uncomment it and replace the word ‘index’ with the word ‘homepage’.

Now reload the browser page and you can see our new sample home page of the Web application.

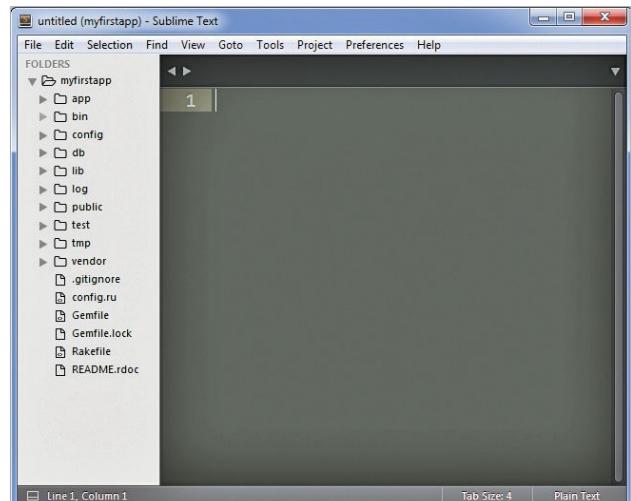


Figure 8: Web application using agency snapshot-1

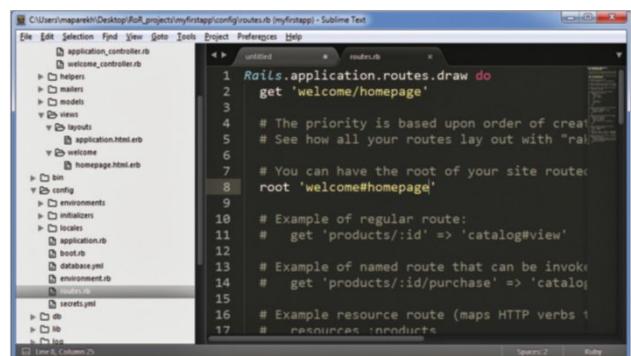


Figure 9: Web application using agency snapshot-2



Figure 10: Web application using agency snapshot-3

We can go to *homepage.html.erb* and write code to create the application.

So this is our sample Web application. Now, we will use one of the bootstrap templates and see how effectively and easily we can build a powerful website and customise it.

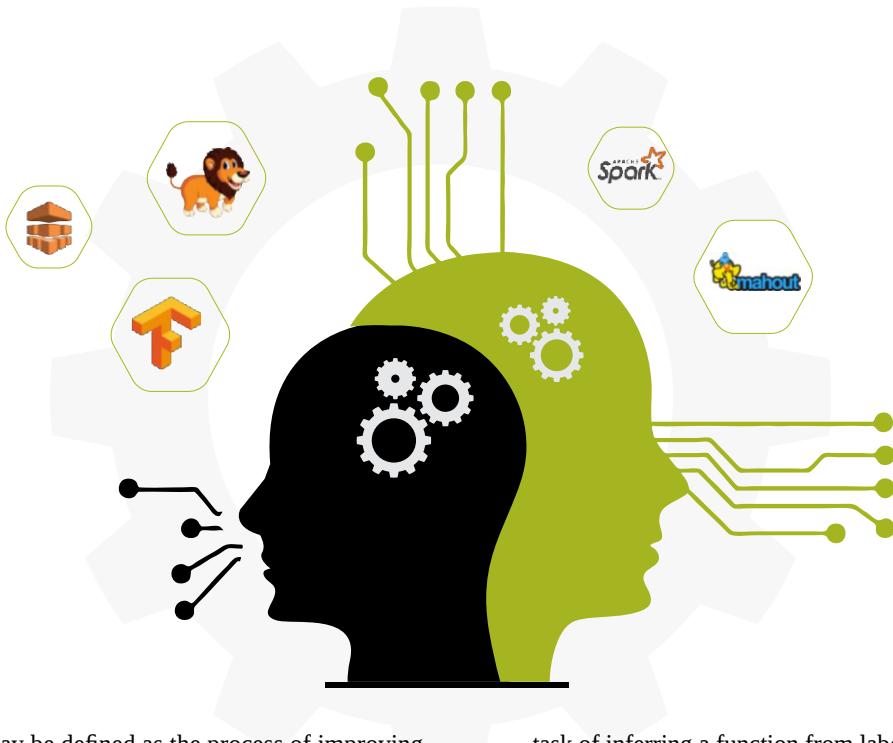
Go to <https://startbootstrap.com/template-overviews/agency/> and download any template—I have chosen the agency template. Download the agency template Zip file and extract it. Compare it with our *myfirstapp* application and add all CSS, JavaScripts and images from the agency template into *myfirstapp/assets/* and the respective folders. Also, copy *index.html* from the template and paste the contents into *homepage.html.erb*.

If your application, running on localhost: 3000, is taking too much time to reload, go to <http://stackoverflow.com/questions/26318837/slow-localhost-load-running-rails-and->

**Continued on Page 71...**

# The Best Open Source Machine Learning Frameworks

In this article, we present what the author rates as the top eight open source machine learning frameworks.



**L**earning may be defined as the process of improving one's ability to perform a task efficiently. Machine learning is another sub-field of computer science, which enables modern computers to learn without being explicitly programmed. Machine learning has basically evolved from artificial intelligence via pattern recognition and computational learning theory. Machine learning explores the area of algorithms, which can make high end predictions on data. In recent times, machine learning has been deployed in a wide range of computing tasks, where designing efficient algorithms and programs becomes rather difficult, such as email spam filtering, optical character recognition, search engine improvement, digital image processing, data mining, etc.

Tom M. Mitchell, renowned computer scientist and professor at Carnegie Mellon University, USA, defined machine learning as: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Machine learning tasks are broadly classified into three categories, depending on the nature of the learning 'signal' or 'feedback' available to a learning system.

- *Supervised learning* is regarded as a machine learning

task of inferring a function from labelled training data. In supervised learning, each example is a pair consisting of an input object (vector) and a desired output value (supervisory signal).

- *Unsupervised learning*: This is regarded as the machine learning task of inferring a function to describe hidden structures from unlabelled data. It is closely related to the problem of density estimation in statistics.
- *Reinforcement learning* is an area of machine learning that is linked to how software agents take actions in the environment so as to maximise some notion of cumulative reward. It is applied to diverse areas like game theory, information theory, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is formulated as a Markov decision process (MDP) due to dynamic programming techniques.

The application of machine learning to diverse areas of computing is gaining popularity rapidly, not only because of cheap and powerful hardware, but also because of the increasing availability of free and open source software, which enable machine learning to be implemented easily. Machine learning practitioners and researchers, being a part of the software engineering team, continuously build sophisticated products, integrating intelligent algorithms

with the final product to make software work more reliably, quickly and without hassles.

There is a wide range of open source machine learning frameworks available in the market, which enable machine learning engineers to build, implement and maintain machine learning systems, generate new projects and create new impactful machine learning systems.

Let's take a look at some of the top open source machine learning frameworks available.

## Apache Singa

The Singa Project was initiated by the DB System Group at the National University of Singapore in 2014, with a primary focus on distributed deep learning by partitioning the model and data onto nodes in a cluster and parallelising the training. Apache Singa provides a simple programming model and works across a cluster of machines. It is primarily used in natural language processing (NLP) and image recognition. A Singa prototype accepted by Apache Incubator in March 2015 provides a flexible architecture of scalable distributed training and is extendable to run over a wide range of hardware.

Apache Singa was designed with an intuitive programming model based on layer abstraction. A wide variety of popular deep learning models are supported, such as feed-forward models like convolutional neural networks (CNN), energy models like Restricted Boltzmann Machine (RBM), and recurrent neural networks (RNN). Based on a flexible architecture, Singa runs various synchronous, asynchronous and hybrid training frameworks.

Singa's software stack has three main components: Core, IO and Model. The Core component is concerned with memory management and tensor operations. IO contains classes for reading and writing data to the disk and the network. Model includes data structures and algorithms for machine learning models.

Its main features are:

- Includes tensor abstraction for strong support for more advanced machine learning models
- Supports device abstraction for running on varied hardware devices
- Makes use of *cmake* for compilation rather than *GNU autotool*
- Improvised Python binding and contains more deep learning models like VGG and ResNet
- Includes enhanced IO classes for reading, writing, encoding and decoding files and data

The latest version is 1.0.

**Website:** <http://singa.apache.org/en/index.html>

## Shogun

Shogun was initiated by Soeren Sonnenburg and Gunnar Raetsch in 1999 and is currently under rapid development by a large team of programmers. This free and open source

toolbox written in C++ provides algorithms and data structures for machine learning problems. Shogun Toolbox provides the use of a toolbox via a unified interface from C++, Python, Octave, R, Java, Lua and C++; and can run on Windows, Linux and even MacOS. Shogun is designed for unified large-scale learning for a broad range of feature types and learning settings, like classification, regression, dimensionality reduction, clustering, etc. It contains a number of exclusive state-of-art algorithms, such as a wealth of efficient SVM implementations, multiple kernel learning, kernel hypothesis testing, Krylov methods, etc.

Shogun supports bindings to other machine learning libraries like LibSVM, LibLinear, SVMLight, LibOCAS, libqp, VowpalWabbit, Tapkee, SLEP, GPML and many more.

Its features include one-time classification, multi-class classification, regression, structured output learning, pre-processing, built-in model selection strategies, visualisation and test frameworks; and semi-supervised, multi-task and large scale learning.

The latest version is 4.1.0.

**Website:** <http://www.shogun-toolbox.org/>

## Apache Mahout

Apache Mahout, being a free and open source project of the Apache Software Foundation, has a goal to develop free distributed or scalable machine learning algorithms for diverse areas like collaborative filtering, clustering and classification. Mahout provides Java libraries and Java collections for various kinds of mathematical operations.

Apache Mahout is implemented on top of Apache Hadoop using the MapReduce paradigm. Once Big Data is stored on the Hadoop Distributed File System (HDFS), Mahout provides the data science tools to automatically find meaningful patterns in these Big Data sets, turning this into 'big information' quickly and easily.

- *Building a recommendation engine:* Mahout provides tools for building a recommendation engine via the Taste library-- a fast and flexible engine for CF.
- *Clustering with Mahout:* Several clustering algorithms are supported by Mahout, like Canopy, k-Means, Mean-Shift, Dirichlet, etc.
- *Categorising content with Mahout:* Mahout uses the simple Map-Reduce-enabled naïve Bayes classifier.

The latest version is 0.12.2.

**Website:** <https://mahout.apache.org/>

## Apache Spark MLLib

Apache Spark MLLib is a machine learning library, the primary objective of which is to make practical machine learning scalable and easy. It comprises common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction as well as lower-level optimisation primitives and higher-level pipeline APIs.

Spark MLlib is regarded as a distributed machine learning framework on top of the Spark Core which, mainly due to the distributed memory-based Spark architecture, is almost nine times as fast as the disk-based implementation used by Apache Mahout.

The various common machine learning and statistical algorithms that have been implemented and included with MLlib are:

- Summary statistics, correlations, hypothesis testing, random data generation
- *Classification and regression:* Supports vector machines, logistic regression, linear regression, naïve Bayes classification
- Collaborative filtering techniques including Alternating Least Squares (ALS)
- Cluster analysis methods including k-means and Latent Dirichlet Allocation (LDA)
- Optimisation algorithms such as stochastic gradient descent and limited-memory BFGS

The latest version is 2.0.1.

**Website:** <http://spark.apache.org/mllib/>

## TensorFlow

TensorFlow is an open source software library for machine learning developed by the Google Brain Team for various sorts of perceptual and language understanding tasks, and to conduct sophisticated research on machine learning and deep neural networks. It is Google Brain's second generation machine learning system and can run on multiple CPUs and GPUs. TensorFlow is deployed in various products of Google like speech recognition, Gmail, Google Photos and even Search.

TensorFlow performs numerical computations using data flow graphs. These elaborate the mathematical computations with a directed graph of nodes and edges. *Nodes* implement mathematical operations and can also represent endpoints to feed in data, push out results or read/write persistent variables. *Edges* describe the input/output relationships between nodes. *Data edges* carry dynamically-sized multi-dimensional data arrays or tensors.

Its features are listed below.

- *Highly flexible:* TensorFlow enables users to write their own higher-level libraries on top of it by using C++ and Python, and express the neural network computation as a data flow graph.
- *Portable:* It can run on varied CPUs or GPUs, and even on mobile computing platforms. It also supports Docker and running via the cloud.
- *Auto-differentiation:* TensorFlow enables the user to define the computational architecture of predictive models combined with objective functions, and can handle complex computations.
- *Diverse language options:* It has an easy Python based interface and enables users to write code, and see visualisations and data flow graphs.

The latest version is 0.10.0.

**Website:** [www.tensorflow.org](http://www.tensorflow.org)

## Oryx 2

Oryx 2 is a realisation of Lambda architecture built on Apache Spark and Apache Kafka for real-time large scale machine learning. It is designed for building applications and includes packaged, end-to-end applications for collaborative filtering, classification, regression and clustering.

Oryx 2 comprises the following three tiers.

- *General Lambda architecture tier:* Provides batch, speed and serving layers, which are not specific to machine learning.
- Specialisation on top which, in turn, provides machine learning abstraction to hyperparameter selection, etc.
- End-to-end implementation of the same standard machine learning algorithms as an application (ALS, random decision forests, k-means) on top.

Oryx 2 consists of the following layers of Lambda architecture as well as connecting elements.

- *Batch layer:* Used for computing new results from historical data and previous results.
- *Speed layer:* Produces and publishes incremental model updates from a stream of new data.
- *Serving layer:* Receives models and updates, and implements a synchronous API, exposing query operations on results.
- *Data transport layer:* Moves data between layers and takes input from external sources.

The latest version is 2.2.1.

**Website:** <http://oryx.io/>

## Accord.NET

Accord.NET is a .NET open source machine learning framework for scientific computing, and consists of multiple libraries for diverse applications like statistical data processing, pattern recognition, linear algebra, artificial neural networks, image and signal processing, etc.

The framework is divided into libraries via the installer, compressed archives and NuGet packages, which include Accord.Math, Accord.Statistics, Accord.MachineLearning, Accord.Neuro, Accord.Imaging, Accord.Audio, Accord.Vision, Accord.Controls, Accord.Controls.Imaging, Accord.Controls.Audio, Accord.Controls.Vision, etc.

Its features are:

- Matrix library for an increase in code reusability, and gradual change of existing algorithms over standard .NET structures.
- Consists of more than 40 different statistical distributions like hidden Markov models and mixture models.
- Consists of more than 30 hypothesis tests like ANOVA, two-sample, multiple-sample, etc.
- Consists of more than 38 kernel functions like KVM, KPC and KDA.

The latest version is 3.1.0.

**Website:** [www.accord-framework.net](http://www.accord-framework.net)

## Amazon Machine Learning (AML)

Amazon Machine Learning (AML) is a machine learning service for developers. It has many visualisation tools and wizards for creating high-end sophisticated and intelligent machine learning models without any need to learn complex ML algorithms and technologies. Via AML, predictions for applications can be obtained using simple APIs without using custom prediction generation code or complex infrastructure.

AML is based on simple, scalable, dynamic and flexible ML technology used by Amazon's 'Internal Scientists' community professionals to create Amazon Cloud Services. AML connects to data stored in Amazon S3, Redshift or RDS, and can run binary classification, multi-class categorisation or regression on this data to create models.

The key contents used in Amazon ML are listed below.

- **Datasources:** Contain metadata associated with data inputs to Amazon ML.
- **ML models:** Generate predictions using the patterns extracted from the input data.
- **Evaluations:** Measure the quality of ML models.
- Batch predictions asynchronously generate predictions for multiple input data observations.
- Real-time predictions synchronously generate predictions for individual data observations.

Its key features are:

## Continued from Page 67...

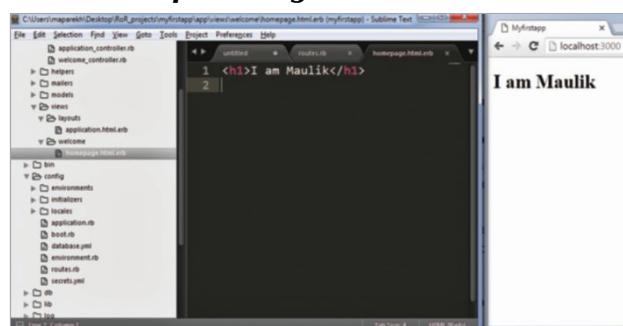


Figure 11: Final Web application using the agency template

webrick and make the changes accordingly.

Now, once all the contents are replaced, you can see the website.

We now have to resolve the following issues:

- 1) All images are not shown in the website; so go to `homepage.html.erb` and search for `src="img"`, then replace `img` with `assets`, and reload the page. You can see all the images, except the background image.
- 2) The background image is not visible, so go to `stylesheets/agency.css` and search for `background-image: url('..img/header-bg.jpg');` then replace `img` with `assets`. The result will look like this background-image: URL

- Supports multiple data sources within its system.
- Allows users to create a data source object from data residing in Amazon Redshift – the data warehouse Platform as a Service.
- Allows users to create a data source object from data stored in the MySQL database.
- Supports three types of models: binary classification, multi-class classification and regression.

**Website:** <https://aws.amazon.com/machine-learning/> 

## References

- [1] <http://singa.apache.org/en/index.html>
- [2] <http://www.shogun-toolbox.org/>
- [3] <https://mahout.apache.org/>
- [4] <http://spark.apache.org/mllib/>
- [5] <http://www.tensorflow.org>
- [6] <http://oryx.io/>
- [7] [www.accord-framework.net](http://www.accord-framework.net)
- [8] <https://aws.amazon.com/machine-learning/>

## By: Prof. Anand Nayyar

The author is assistant professor in the department of computer applications and IT at KCL Institute of Management and Technology, Jalandhar, Punjab. He loves to work on and research open source technologies, cloud computing, sensor networks, hacking and network security. He can be reached at [anand\\_nayyar@yahoo.co.in](mailto:anand_nayyar@yahoo.co.in). Watch his YouTube videos at [Youtube.com/anandnayyar](https://www.youtube.com/anandnayyar).

(`..../assets/header-bg.jpg`);

- 3) As this template uses the font 'Awesome', different font icons are not visible on the page. Please follow <http://stackoverflow.com/questions/11052398/rails-using-fontawesome> to solve this problem.

Finally, you can see the entire website running, as shown in Figure 11. 

## References

- [1] <http://rubyonrails.org/>
- [2] [http://installrails.com/steps/choose\\_os](http://installrails.com/steps/choose_os)
- [3] **System path is not found issue**  
<https://github.com/railsinstaller/railsinstaller-windows/issues/73>
- [4] <https://startbootstrap.com/template-overviews>
- [5] **Local server is running slow issue**  
<http://stackoverflow.com/questions/26318837/slow-localhost-load-running-rails-and-webrick>

## By: Maulik Parekh

The author has an M. Tech degree in cloud computing from VIT University, Chennai. He has rich and varied experience at various reputed IT organisations, and can be reached at [maulikparekh2@gmail.com](mailto:maulikparekh2@gmail.com) or <https://www.linkedin.com/in/maulikparekh2>.

# Autolt: An Open Source Software Testing Tool for Windows

The official website of AutoIt defines it as a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. AutoIt runs on all versions of Windows. This third-generation programming language can be used to automate any specific task by simply writing a few lines of script.



It can hardly be denied that many of us are becoming a bit lazy about performing tasks manually, and try our best to make the system do all repetitive tasks to the highest extent possible. If you are running out of time for completing a specific task on some online application within a deadline, you will rely on the system to do so with accuracy, reducing the chance of last minute errors that might creep in if the task is done manually. That is why automation is so important in the current technology-driven era.

Automation testing gauges the quality of software with other tools or software code, which compare the actual outcomes with the predicted outcomes. The entire test suite is run without any manual intervention. The tool itself generates the final outcome of the testing performed in the form of a test report, which features the expected and actual outcomes. The specific method or process that is used to implement the automated process is called a test automation framework. The tool can even help us log defects if required. Automation testing is one of the most happening areas of the software quality assurance industry, as most organisations prefer to rid themselves of manual testing as much as possible. They want to rely on the system or on some tool. Let's have a look at the possible reasons for this.

- Manual testing is time- and cost-consuming when it comes to covering all workflows, all fields and

all negative scenarios. If the same set of validations is performed at different intervals, as in the case of regression testing, then automation testing is quite important.

- Automation testing does not require any human intervention. The automated test suite can be run unattended (even overnight).
  - It increases the speed of test execution.
  - It increases test coverage.
  - Manual testing can also become boring at times and, therefore, it is error prone.

## Some open source automation testing tools

There are many options to test any type of software. Some tools are available as open source while some are licensed, for which one has to pay before use. The first choice of any organisation is always for open source tools as they are free of cost. But there are certain limitations associated with these tools, due to which some firms opt for licensed tools. Table 1 lists some open source tools that are being used to test various types of software applications.

## The AutoIt tool

AutoIt is an open source tool that can be used to automate various processes involving Windows and desktop

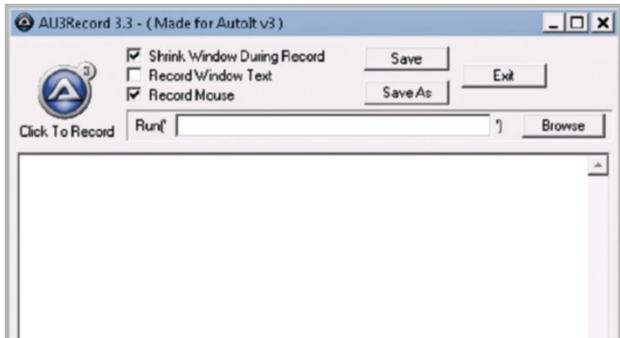


Figure 1: AutoIt recorder window

applications. The official website of AutoIt defines it as a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It basically uses a combination of different keystrokes, mouse movements and window/control manipulation to automate different tasks. AutoIt runs on all versions of Windows. Since its initial release around 17 years ago by Jonathan Bennett and the AutoIt team in January 1999, it has become a very powerful language that supports various complex expressions, user functions, loops and other such features.

The syntax of the scripting language used in AutoIt is similar to that found in the BASIC family of languages. It actually uses C# and VB as its scripting language. AutoIt is a third-generation

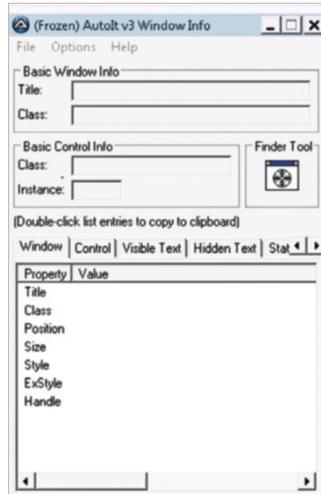


Figure 2: AU3 Info tool

programming language with a classical data model and various data types which can store several types of data. It can be used to automate any specific task by simply writing a few lines of script or even record that process using the AutoIt Recorder. Once the recording is done, the script is automatically generated and we can use the same to further play back the recorded process. It identifies several elements of the application by using their various properties like the control ID, coordinates,

the element's name and other such smart identifiers. All such identifiers' values are stored in a repository in the form of objects, and they are used during playback to identify various elements and perform actions on them.

The AU3 Info object identifier helps retrieve these values for a specific element of an application. It also has several inbuilt function libraries containing predefined functions that must be imported so that different functions can be used in the script. It has a SciTE editor, which

Table 1: A few open source automation testing tools'

| Tool name | Type of applications tested           | Supported scripting language              | Other features                                                                                                                                                                                                                                                                                                                                          |
|-----------|---------------------------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Selenium  | Web applications                      | Java, C#, Ruby, Groovy, PERL, Python, PHP | <ul style="list-style-type: none"> <li>Provides record/playback tool to automate processes</li> <li>Can be deployed on Windows, Linux and Macintosh platforms</li> <li>Identifies components of the application using various element properties</li> <li>Consists of the Selenium IDE, client API, Web driver, grid and remote control</li> </ul>      |
| AutoIt    | Windows and desktop applications      | VB and C#                                 | <ul style="list-style-type: none"> <li>Provides record/playback option to automate processes</li> <li>Consists of various inbuilt library functions to perform specific tasks</li> <li>Uses control ID, position coordinates and other such properties to identify elements</li> <li>Keyboard shortcuts can be easily implemented using this</li> </ul> |
| Sikuli    | Windows, desktop and Web applications | Python, Java and Ruby                     | <ul style="list-style-type: none"> <li>Uses image recognition to recognise and control GUI elements</li> <li>Supports scripting with SikuliXIDE</li> </ul>                                                                                                                                                                                              |
| Sahi      | Web applications                      | JavaScript                                | <ul style="list-style-type: none"> <li>Has a record and playback feature to automate various processes</li> <li>Uses Smart Accessor Identification to identify elements with even dynamic IDs</li> <li>Consists of rich inbuilt reports and logs</li> <li>Fast parallel playback feature, which can run several scripts at a time</li> </ul>            |
| SoapUI    | Web applications                      | Groovy script and Java                    | <ul style="list-style-type: none"> <li>Useful for functional Web services, security and load testing</li> <li>Supports data driven testing</li> <li>Supports multiple environments</li> <li>Generates quick reports</li> <li>Provides drag-and-drop option for creating test suites, test steps, etc</li> </ul>                                         |



Figure 3: AutoIt GUI icon selector

helps to build and run various programs. The SciTE editor also provides all the configuration settings and other utility programs like Code Folding, Auto-complete, Syntax Highlighting and Intellisense. AutoIt is distributed with an IDE which is based on the free SciTE editor. The compiler and its help text are fully integrated and provide a standard environment for developers. We can create various GUIs such as Input Box, Label, Browser window, etc, which can be used to get a feel of the built application while creating a script. Once we have created a script, we can save it in the form of a .au3 executable file. It can be converted into a compressed and standalone executable, which can even be run on computers that do not have the AutoIt interpreter installed.

## Features of AutoIt

Let's check the various features of AutoIt.

**Easy to learn syntax:** AutoIt uses a very simple syntax for its scripting language, which can be easily learnt and implemented for the automation of any process. The SciTE editor makes it easy by providing the syntax highlighting feature, using which one can get an idea of the syntax of the various predefined functions used.

**Record and playback:** As previously discussed, AutoIt can even record and generate a script on its own for specific processes that need to be automated. Once recorded, the same script, without any modification, can be used to run and play back the recorded process when required.

**Interaction with all standard Windows controls:** AutoIt can easily interact with all types of standard Windows controls and other GUIs without any object identification issue. It identifies a specific control with the help of the control ID, class name, title, position coordinates, visible text, etc – all of which are associated with that Windows control. It can get such properties of an element using the AU3 Info tool.

**Simulates keystrokes and various mouse movements:** AutoIt can even simulate various mouse movements and different keystrokes using the inbuilt 'Send Key' function. These movements and different key actions can also be recorded and played back afterwards.

**Graphical User Interfaces (GUIs):** Different GUIs can

be provided to different processes, wherever required while automating a specific task, using AutoIt. There are several inbuilt functions under 'GUI Control Create' for different GUIs, which can be incorporated. Each of the separate GUI functions has a different syntax, following which we can generate different GUIs like the input box, label, browser window, drop-down menu, etc.

### Calling external DLL and Windows API functions

**directly:** We can directly call any DLL using AutoIt with the help of its predefined function, DLLcall. We can also call any API using this, and can even perform different operations on the Windows API using several \_WinAPI\_ functions.

**RunAs function:** AutoIt has an inbuilt RunAs function, which helps in running any external program using some external user.

**Detailed 'Help' file and large community-based support forums:** AutoIt comes with AutoIt Help.exe, which provides all the possible help and support information for any learner. It even illustrates the automation of several scenarios, where the tool was used.

**Compatibility:** AutoIt is compatible with all Windows versions — Windows XP, 2003, Vista and 2008; Windows 7, 2008 R2, Windows 8 and 2012 R2.

**Compilation of scripts into standalone executables:** The entire script written using AutoIt can be stored in the form of a .au3 executable, which can be easily compiled and executed by simply clicking on the .au3 executable file. We need not open the script to run it.

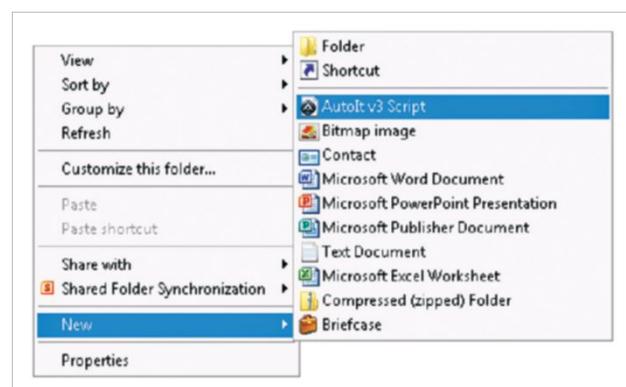


Figure 4: Creating a new .au3 file

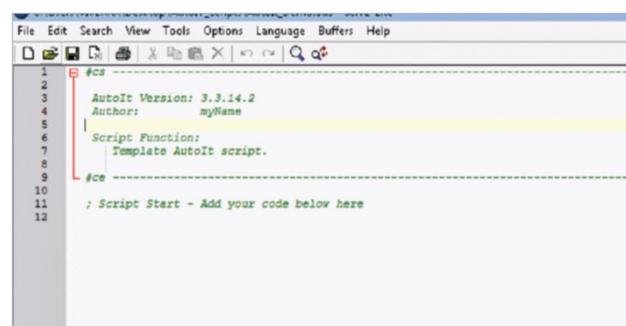


Figure 5: AutoIt script editor window

## How to automate a process using AutoIt

We first need to have AutoIt installed on our system. It just takes a few minutes to install it—simply click on *autoit setup.exe*. We need to additionally install *Scite Editor.exe* as well, in order to have the script editor in AutoIt.

Once we are done with the installation part, we need to simply right-click anywhere on the location where AutoIt has been installed, and follow the steps shown below to get the new *.au3* file created on our system.

Right click *New -> AutoIt v3 script* -> Give a name to the file -> Press *Enter*.

Once we have created the *.au3* file using the above steps, we need to just right click on this file and select the *Edit script* option to get an AutoIt script editor window, as shown in Figure 5.

This is where we can try our hand at scripting in order to make the AutoIt tool perform some task or action. Let's automate a simple scenario where the user is asked his/her name, and once that is entered in the input box, the user gets a 'Welcome' message. Before proceeding with the scripting part, we can add our name as the *Author* in the auto-generated template of the script editor window.

The requirements are:

1. The ability to display the question asking the user for his/her user name and to accept this input. To incorporate this, we need to have the predefined *InputBox* function of AutoIt.
2. The inbuilt Message Box function to display a *Welcome* message for the user. We will have to import the *MessageBoxConstants* library to use the Message Box function in our script.

Let's start scripting with the following steps:

1. Import all the required library files -- we need to include the *MessageBoxConstants* library file.
2. Declare all the types of variables required (integer, string, etc) using the keyword *Dim*.
3. As discussed in Requirement 1, use the *InputBox* function to ask and take the input from the user. We just need to type *Input* and then click *Ctrl + Space* to get various hints, and then we can select *InputBox* out of the options provided.
4. Insert an opening round brace just after *InputBox*; AutoIt will display the hint of the syntax to be followed for that specific function. So here, for *InputBox*, we need to enter the following parameters in sequence.
  - Title – title to appear on top of the Input Box window
  - Prompt – the question for which the user needs to provide input.
5. Once the user enters his/her name, we need to display the *Welcome* message using the name; so we will store the value of *InputBox* in a variable. We also need to verify whether the user has entered a name or not, so that we can display the *Welcome* or *Error* message accordingly. Hence, we will use the conditional *If-else* for that.

Finally, we are ready with the following AutoIt code:

```
; Including library files
#include <MsgBoxConstants.au3>

; Variable Declaration
Dim $UserName

; Taking input from User
$UserName=InputBox("User Name","Please enter the User Name")

; Validating if User has entered some User Name and then
; displaying Welcome Message
If $UserName <> "" Then
 MsgBox(0,"Welcome Message","Hey " & $UserName & ", Welcome to
 AutoIt..!!!")
Else
 MsgBox(0,"Error Message", "Please enter the User Name")
EndIf
```

Once we run the above code using F5, we will get an input box for the user to enter his/her name (Figure 6).

Once the user enters some name, a *Welcome* message will be displayed (Figure 7).

If the user does not enter any name and clicks on *OK*, then an *Error* message is generated.



Figure 6: Input box window

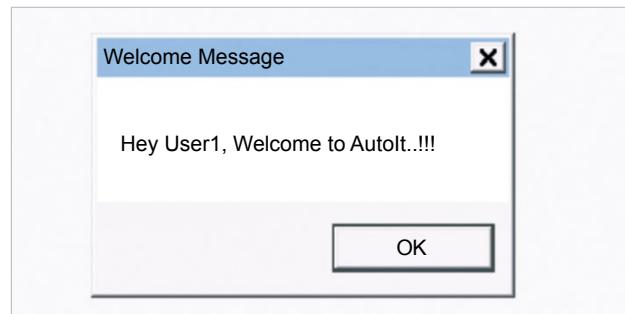


Figure 7: Welcome message box



Figure 8: Error message box

## Advantages of AutoIt

1. Supports easy record/playback and simple scripting
2. Loglines can be debugged easily with `consolewritecommand`.
3. AutoIt is an open source tool for which we do not need to pay.
4. Small standalone executables can be created using AutoIt.
5. AutoIt can recognise specific areas with `pixelchecksum` and `pixelsearch`.
6. It recognises all basic Windows controls and also has the option to create different GUIs.

## Disadvantages of AutoIt

1. All types of objects are not identified by the AU3 Info tool.

2. AutoIt has no Java support available so far.
3. It has no control repository to give an object a logical or technical name.
4. It does not support all browsers, but only supports Internet Explorer controls by using `IE.AU3` or its own COM object references.
5. Users need to import separate libraries for using different predefined utility functions.

There are several open source tools available in the market, but we need to evaluate and analyse any specific tool based on our requirements. AutoIt is one tool which can be used to automate various processes associated with Windows and desktop based applications with simple, user-friendly scripting. [END](#) 

## References

- [1] <http://www.wikipedia.org/>
- [2] [www.opensourcetesting.org/](http://www.opensourcetesting.org/)
- [3] <https://www.autoitscript.com>

## By: Vivek Ratan

The author is a B. Tech in electronics and instrumentation engineering. He currently works as an automation test engineer at Infosys, Pune. He can be reached at [ratanvivek14@gmail.com](mailto:ratanvivek14@gmail.com)

Where do  
**800,000+**  
electronics design  
enthusiasts meet  
every month?

An  
**EFY GROUP**  
ONLINE PROPERTY



### What about you?

World's leading social media  
platform for electronics design  
professionals & enthusiasts!

[facebook.com/designelectronics](https://facebook.com/designelectronics)



[/designelectronics](https://facebook.com/designelectronics)



# Does Your Mobile App Work Without an Internet Connection?

Native mobile apps act as the interface between the user and the users' data and require uninterrupted connectivity. A native app with offline capability stores both the mobile app's software and its data locally on the mobile device. Offline mobile apps allow the user to run the app, regardless of connectivity.

**H**ow many times have you opened a mobile app, filled out a lengthy form that runs across multiple screens, and passed through validation errors only to see a message stating, "No Internet connection" when you finally click *Submit*? Most mobile applications are simply clients that send and receive data in real-time and become unusable without an active Internet connection.

As per the ICT Facts and Figures 2016, only 40.9 per cent of the population in developing countries has a mobile broadband subscription. And those who have it often complain about fluctuations, especially in crowded places or while travelling. Even in metro cities, the continuity of the network is not perfect. With unreliable networks, limited bandwidth and high latency, it becomes really challenging for engineering teams to come up with solutions that enable users to use an app, regardless of network connectivity. An approach where you make a network request, wait for the response and then display it, is quite unappealing. Ideally, you should have the data already, which you can display immediately, with a separate mechanism to update it.

## Adding the offline capability to apps

Here are a couple of questions that must be answered before taking the plunge. What are the key features of the app that render it unusable due to no/flaky network conditions? How many users are affected due to this? How does it affect the key metrics of your business? Answers to these will help reveal the areas where there is an opportunity to provide offline support. There may not be a need to take the complete app offline. Features that cannot work offline can be redesigned, disabled or even be hidden.

## Offline views

In order for the app to function in unsteady and offline environments, the data that the application needs must be persisted locally on the device for subsequent lookups.

## Caching

There could be different strategies for caching and refreshing



the data. Choosing the right one is important and depends on the type of data your app presents. For example, time-sensitive data like stock updates should be as recent as possible, while images or lists of locations can be updated less frequently.

**Network first:** Try to retrieve data from the server first, and if that is not possible because there is no network, retrieve data from the local cache (if available). The next successful network hit should update the local copy. This strategy is useful when you always want to show the latest and most updated information.

**Local first:** Retrieve data from the cache and display it immediately. At the same time, if the network is available, fetch data from the server in the background and update the view as well as the cache. This approach is very useful as it leads to minimum in-app latency and the user gets to see the data instantly.

**Local only:** Cache data for a specific period and fetch it only from the cache without contacting the server at all. The cache can be later updated via notifications or periodic service polls. Do take into account the risk of battery drain caused by aggressive polling.

**Prefetch:** In some cases, it makes sense to download lightweight content and cache it pre-emptively so that it is instantly available when requested later.

Usually, a combination of these methods is required for different types of data at different places in the app. For example, product details can be cached for long, using either the 'local first' or the 'local only' strategy, whereas product price can be fetched in real-time by always using the 'network first' strategy. The longer the data can be cached, the better it is. Just be conscious of the size and security of data being stored. All sensitive data should be encrypted and stored safely.

## API support

Although the clients can decide on, choose and implement their own caching policy, the complete benefit of caching can be realised only when complemented with some sort of server-side caching support as well. By sending headers like cache-control with appropriate values, the server can indicate

which data to cache and for how long. Networking libraries usually have inbuilt support for such headers and cache responses automatically. The backend can also be designed to reduce the amount of data transfer by comparing the version numbers (or Etags) of resources and sending a representation only if the resource has changed.

## Offline updates

Data once cached can be displayed in offline mode easily, but what about adding/editing data? You may want to allow users to register, post pictures and update profiles, even when offline.

## Queuing

Whenever the application does not have a network connection, the newly entered or edited data operations can be collected in a local queue and processed later. A user should be able to use the app normally and always see the updated data, even if it is unprocessed. The queue also helps maintain the correct sequence of actions, which could be crucial at times. Imagine how confusing a conversation would be if the messages arrived in a random order.

A key point is to always inform users about queued operations. It would be better to create a separate UI that displays a list of unprocessed changes, giving users an option to cancel or retry them. As the operation is performed, notify users about the respective success or failure outcomes. This will help build more trust in the app among users—an assurance that it won't lose any data regardless of the connectivity state.

## Synchronisation

When the signal is restored, the collected changes must be synchronised with data in the backend. Synchronisation can be triggered in various ways—as soon as the network is available, on launching the application, once in a day, after every six hours, etc. It's better to go with bundled data transfers to help conserve battery resources.

## Sync conflicts

While the user was working offline, it is possible that changes were made by other users to the same data. Or, the same user can change his data from multiple sources/devices, which can lead to conflicts that are usually detected at the level of a row in a database. A row is in conflict if it is changed at more than one of the sources between synchronisations. It could be a unique key collision, in which a row with the same unique key is inserted from different sources, or an update collision when the same row is updated from different places. It could also be a delete conflict when updating a row that has been deleted already.

Let's look at an example.

- Clients A and B synchronise with the server and pull version v1 of a resource. The resource can be represented as a row in the database.

- A updates the resource and synchronises with the server. There is no conflict. The version is updated to v2.
- Later, B updates the same resource (at v1) and synchronises. A write conflict occurs because it is an update on the older version and the resource has been updated to v2 already.

Sometimes, conflicts can be detected on the client side.

In the example above, if before pushing the changes, B pulls the resource again, it can compare the versions and see there is a conflict. B can then choose to discard its local changes or apply some other resolution, and inform the user accordingly.

Just like the version control systems, these conflicts must be resolved before the synchronisation process gets completed. There can be different conflict resolution strategies and choosing the right one is important to prevent data inconsistency problems.

**Server wins:** In this approach, changes received during synchronisation are discarded, leaving data in the database unchanged.

**Client wins:** The conflict is ignored and changes received are accepted, overwriting the value in the database. One should be very careful when using this approach as it might lead to data inaccuracy. Incorrectly overwriting a change of location (from Delhi to Mumbai, for instance) will result in showing completely irrelevant recommendations to the user.

**Last update wins:** Data with the most recent modified timestamp wins.

**Let the user decide:** Leave the decision to app users. The users can view the conflicting values and decide which one to keep.

**Custom algorithm:** A custom algorithm defined on the basis of business rules can be used to resolve conflicts. For example, there could be a business rule that says, "Pick the largest of the two values or take the union of the conflicting data sets and merge them."

You may encounter situations where data is complex. Depending on your implementation, you can choose different approaches and build a robust conflict resolution system.

Synchronising offline data makes the application more responsive and optimised. You can build your own system for managing data transfers or use frameworks like Android's Sync Adapter that automate it for you.

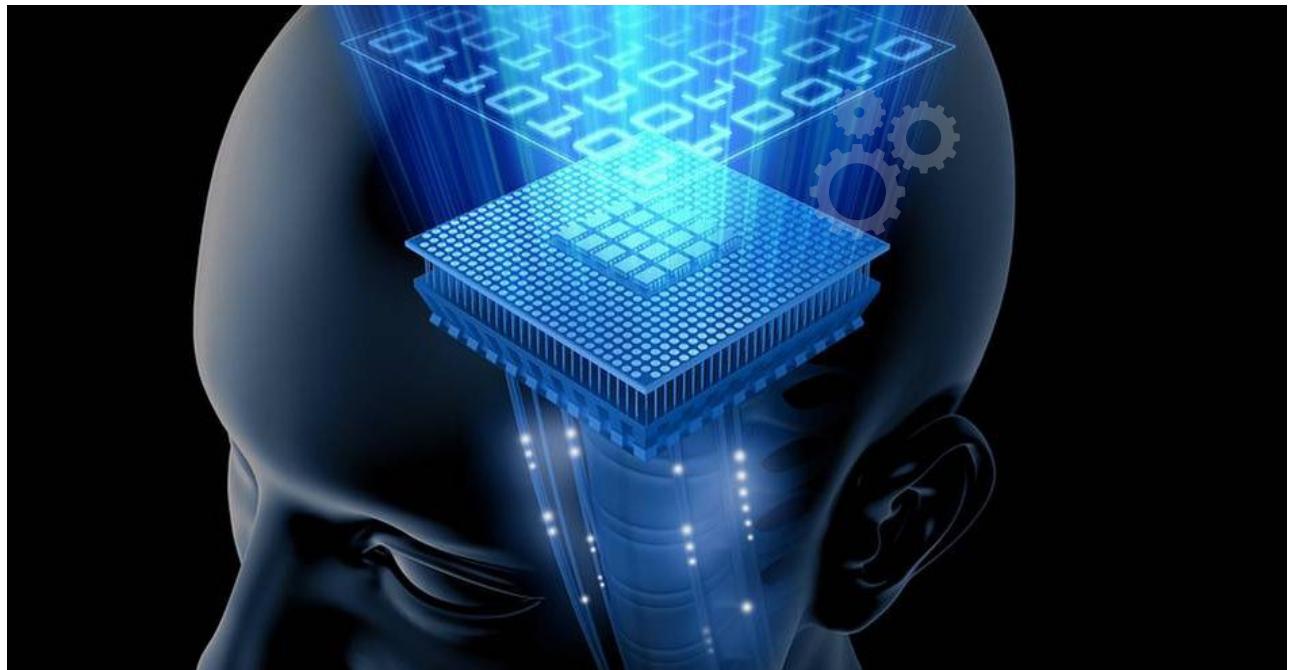
Lack of connectivity is not an error condition. Instead of showing an error message, the app should be tailored to provide a seamless experience, always. After all, we all know a happy customer is the best customer! 

**By: Minni Arora**

The author works as a technical architect with Naukri.com. She has nine years of experience in architecture, design and development of web as well as mobile applications. She can be contacted at [minienarula@gmail.com](mailto:minienarula@gmail.com).

# Applying Machine Learning to Real Business Problems

This article is intended to help developers who are working on machine learning initiatives or are planning to set out on this path. It is important to know about some basic concepts and steps when it comes to applying machine learning to real business problems.



**M**achine learning is a field of artificial intelligence (AI), in which systems are programmed to gather insights from historic data and generate outcomes that can be used for decision making. Machine learning problems can be approached in multiple ways, one of which is collecting lots of data and training the model. We can also derive features from the data and use them to train the model, or pre-process data and transform it into specific formats before using it to train a model. It is not easy to tell which approach will work and give the desired results. Hence, experts suggest some guidelines to approaching a machine learning problem:

- Start with a simple algorithm, then implement and test it with a small data set.
- Plot learning curves to see how it works, and if required, use more data or features.
- Perform error analysis to analyse errors and try to identify a pattern that can help in improving the algorithm.

It is not always recommended that you start with programming a machine learning algorithm but, instead, try it using open source tools such as R or Octave to play around and select the right model that works for particular

types of data.

The objective of this article is to highlight some aspects that are important when selecting an algorithm, setting parameters and diagnosing performance. Some fundamental terms which are referred to in the article are explained below.

**Hypothesis:** A function that fits the data in linear or non-linear fashion

$$\hat{y} = h_{\theta}(x) = \theta_0 + \theta_1 x$$

Here, the hypothesis function is the equation line; so data will be fitted in a line. We give  $h_{\theta}(x)$  values for parameters  $\theta_0$  and  $\theta_1$  to get our estimated output  $\hat{y}$ .

**Cost function:** This is a function that is used to measure the accuracy of the hypothesis function, i.e., how well it fits the data. The function can be described as follows:

$$J(\theta_0, \theta_1) = \left( \frac{1}{2m} \right) \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \left( \frac{1}{2m} \right) \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2, \text{ for } i=1 \text{ to } m$$

To simplify this, it is  $\frac{1}{2} \bar{X}$  where  $\bar{X}$  is the mean of the squares of  $h_{\theta}(x_i) - y_i$ , or the difference between the predicted value and the actual value.

**Gradient descent:** This is a technique to estimate the parameters  $\theta_0$  and  $\theta_1$  of the hypothesis. The gradient descent equation for linear regression using a single variable is given below:

```
repeat until convergence: {
 $\theta_0 := \theta_0 - \alpha (1/m) \sum (h_{\theta}(x_i) - y_i)$, for i=1 to m
 $\theta_1 := \theta_1 - \alpha (1/m) \sum ((h_{\theta}(x_i) - y_i) x_i)$, for i=1 to m
}
```

...where, m is the size of the training set and  $x_i$ ,  $y_i$  are values of the given training set (data).

**Regularisation:** This keeps all the features but reduces the weightage.

$$\min_{\theta} \frac{1}{2}m \left[ \sum (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum (\theta_j)^2 \right], \text{ for } i=1 \text{ to } m \text{ and } j=1 \text{ to } n$$

$\lambda$ , or lambda, is the regularisation parameter. It determines how much the cost of the  $\theta$  parameters is inflated in order to reduce the weightage of the feature in the hypothesis function.

## Evaluating a hypothesis

There is a misconception that the algorithm that produces low errors is also accurate. This may not always be the case, and the main reason for this could be overfitting. Overfitting (high variance) is caused by a hypothesis function that fits the available data well but does not generalise to predict new data. So it is very important to understand whether the algorithm is overfitting or underfitting (high bias), when the hypothesis function ( $h$ ) maps poorly to the trend of the data. The following section provides some general guidelines to evaluate various aspects of the hypothesis/algorithm.

**Model selection:** Model selection is the process by which we choose the hypothesis with the least errors (more generalised), set parameters and identify any pre-processing requirement to predict the best results. However, it has been observed that errors will always be low on the data used for training the model compared to any other data. Hence, experts recommend another approach using a cross-validation set as detailed below.

- Split the dataset into three sets—a training set, a test set and a cross-validation set. The possible breakdown could be 60-20-20.
- Optimise the parameters in  $\theta$  using the training set.
- Find the polynomial degree  $d$  with the least error using the cross-validation set.
- Estimate the generalisation error using the test set.

In this way, the degree of the polynomial ( $d$ ) has not been trained using the test set and, hence, the generalisation error can be minimised.

**Diagnosis of bias vs variance:** The study of relationships between the cost function and the degree of the polynomial ( $d$ ) can help us determine whether underfitting or overfitting is the problem with our hypothesis. Here are some trends to look

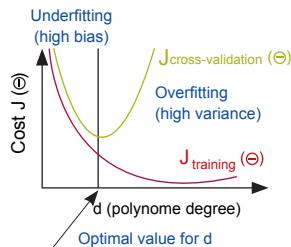


Figure 1: Diagnosis of bias + variance

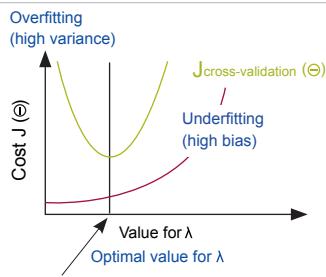


Figure 2: Regularisation and bias + variance

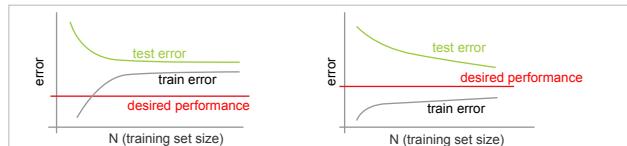


Figure 3: Learning curves

for during the diagnosis of bias and variance.

- Errors on training data decrease when we increase the degree of polynomial ( $d$ ).
- Cross-validation errors decrease on increasing ( $d$ ) up to a certain point, and then start to increase again.

Figure 1 summarises these trends.

Underfitting happens when ( $d$ ) is small, and errors on both training data and cross-validation data are high. Overfitting happens when ( $d$ ) is large, and training data errors are low while cross-validation errors are comparatively higher.

In case of regularisation, we can use the regularisation parameter  $\lambda$  to determine high bias/variance. The rule of thumb to use is:

- Large  $\lambda$ : High bias (underfitting)
- Intermediate  $\lambda$ : Just right
- Small  $\lambda$ : High variance (overfitting)

The relationship of  $\lambda$  to the training and cross-validation errors is shown in Figure 2.

Underfitting happens when  $\lambda$  is large, and errors on both training and cross-validation data are high.

Overfitting happens when  $\lambda$  is small, and errors on training data are low while cross-validation errors are high.

**Learning curves:** Learning curves can help in choosing the best value for ( $d$ ) or  $\lambda$  to plot them. They can help us understand the relationship between errors and the test data size in situations of high bias/variance.

In case of a hypothesis with high bias

(see the left of Figure 3):

- Low data size causes training errors to be low and cross-validation errors to be high.
- Large data size causes both training and cross-validation errors to be high (very similar) and become stable.

This means that the algorithm suffers from high bias.

So getting more training data will not be very useful for further improvement.

In case of a hypothesis with high variance

(see the right of Figure 3):

- Low data size causes training errors to be low and cross-validation errors to be high.
- Large data size causes training errors to increase and cross-validation errors to decrease (without stabilising).

This means that the algorithm suffers from high variance, so getting more training data may be useful.

**Error metrics for skewed classes:** The data sets is largely skewed when we have many more samples of one class compared to other classes in the entire data sets. In such a situation, we can use two attributes of the algorithm, called *Precision* and *Recall*, to evaluate it. In order to determine these attributes, we need to use the following truth table.

|                | PREDICTED | ACTUAL |
|----------------|-----------|--------|
| TRUE POSITIVE  | 1         | 1      |
| TRUE NEGATIVE  | 0         | 0      |
| FALSE NEGATIVE | 0         | 1      |
| FALSE POSITIVE | 1         | 0      |

So,

$$\text{Precision (P)} = \frac{\text{True Positives}}{\text{Total Number of Predicted Positives}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

and

$$\text{Recall (R)} = \frac{\text{True Positives}}{\text{Total Number of Actual Positives}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

We need both *Precision* and *Recall* to be high for the algorithm to perform better. Instead of using these separately, there is a simpler solution—combining them in a single value called F1 score. The formula to calculate the F1 score is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

We need a high F1 score (so both P and R should be large). Also, the F1 score should be obtained over the cross-validation data set.

A few rules for running diagnostics are:

- More training examples fix high variance but not high bias.
- Fewer features fix high variance but not high bias.
- Additional features fix high bias but not high variance.

- The addition of polynomial and interaction features fixes high bias but not high variance.
- When using gradient descent, decreasing lambda can fix high bias and increasing lambda can fix high variance (lambda is the regularisation parameter).
- When using neural networks, small neural networks are more prone to underfitting and big neural networks are prone to overfitting. Cross-validation of the network size is a way to choose alternatives.

## Tips on using the most common algorithms

**SVM:** SVM is one of the most powerful supervised algorithms. Listed below are some tips to applying the SVM algorithm:

- If the feature set ( $n$ ) is large relative to the data sets size ( $m$ ), then use SVM with logistic regression or linear kernel. We don't have enough examples, which would have required a complicated polynomial hypothesis.
- If  $n$  is small and  $m$  is intermediate, then use SVM with a Gaussian Kernel. We have enough examples so we may need a complex non-linear hypothesis.
- If  $n$  is small and  $m$  is large, then manually create/add more features, and use logistic regression or SVM without a kernel. Increase the features so that logistic regression becomes applicable

**K-Means:** K-Means is most popular and a widely used clustering algorithm for unsupervised data sets. Some tips on applying K-Means are listed below:

- Make sure the number of your clusters is less than the number of your training examples.
- Randomly pick K training examples (be sure the selected examples are unique). Determine K by plotting it against the cost function. The cost function should reduce as we increase the number of clusters, and then flatten out. Choose K at the point where the cost function starts to flatten out.

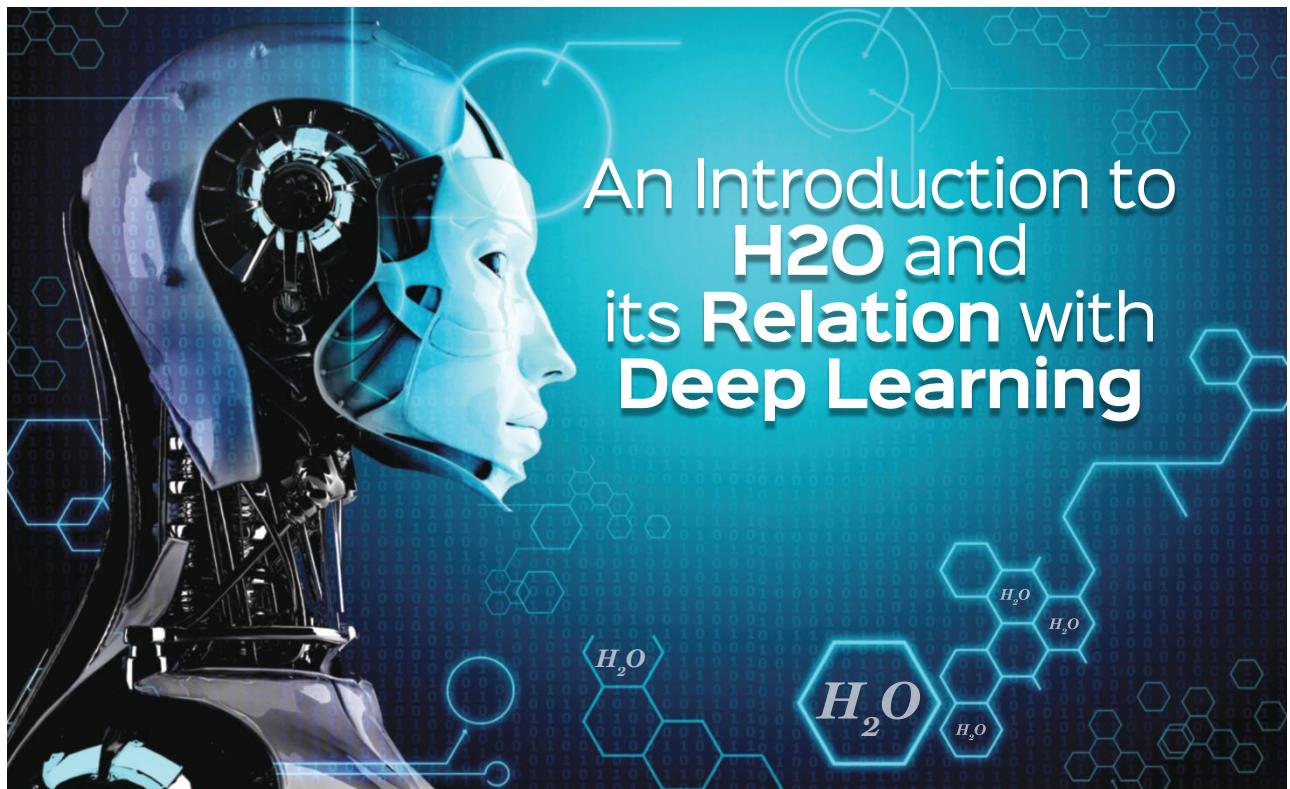
The availability of so many machine learning libraries today makes it possible to start building self-learning applications. However, when we talk about applying machine learning algorithms to real business problems, they need to be highly generalised, scalable and accurate in their outcome – be it prediction or clustering. This is not a very easy task, and requires a process involving various steps to make the algorithms work in the desired fashion. 

## References

This article is largely inspired by a machine learning course by Andrew Ng on Coursera.

## By: Amit Badheka

The author is currently working as a senior technical architect with Capgemini India Pvt Ltd. He is a TOGAF certified enterprise architect with 17 years of IT experience in various capabilities. His areas of interest include various emerging technologies such as IoT, predictive analytics and cognitive computing. He can be reached at [amit.badheka@capgemini.com](mailto:amit.badheka@capgemini.com).



H2O, produced by the start up H2O.ai (formerly Oxdata), was launched in Silicon Valley in 2011. The speed of this open source software for big data analytics allows users to fit thousands of models for pattern discovery in data. H2O is written in Java, Python and R, and has many useful features on offer for deep learning.

**D**eep learning, defined simply, is 'a class of machine learning techniques, where many layers of information processing stages in hierarchical supervised architectures are exploited for unsupervised feature learning and for pattern analysis/classification'. The essence of deep learning is to compute hierarchical features or representations of observational data, where higher-level features or factors are defined from lower-level ones. Although there are many similar definitions and architectures for deep learning, two common elements in all of them are: multiple layers of non-linear information-processing, and supervised or unsupervised learning of feature representations at each layer, from the features learned at the previous layer. The initial work on deep learning was based on multi-layer neural network models. Recently, many other models are also being used, such as deep kernel machines and deep Q-networks.

## Introducing H2O

According to Oxdata (its creator), H2O is 'the open source in-memory, prediction engine for Big Data science'. H2O is a feature-rich, open source machine learning platform known for its R and Spark integration and its ease of use. It is a Java virtual machine that is optimised for doing in-memory processing of distributed, parallel machine learning

algorithms on clusters. A cluster is a software construct that can be fired up on your laptop, on a server, or across the multiple nodes of a cluster of real machines, including computers that form a Hadoop cluster.

## Deep learning for IoT using H2O

In a nutshell, deep learning methods implement neural network algorithms such as the Feed Forward Back Propagation Neural Network, Convolved Neural Networks, Recursive Neural Networks and others.

Deep learning algorithms play an important role in IoT analytics. Data from machines is sparse and/or has a temporal element in it. Even when we trust data from a specific device, it may behave differently under different conditions. Hence, capturing all scenarios for the data pre-processing/training stage of an algorithm is difficult. Monitoring sensor data continuously is also cumbersome and expensive. Deep learning algorithms can help to mitigate these issues. These algorithms learn on their own, allowing the developer to concentrate on more important tasks without worrying about training them.

Let's now discuss what H2O offers as a part of its deep learning framework, and the features that make it suitable for data from things.

## Configuring and loading data in a H2O cluster from R

To boot H2O up in R from a local host, you should check the code. *h2o.init()* provides the method to specify the IP, port and the number of threads to be used by H2O. By default, it uses all threads available in that machine. The first step is to start an instance of H2O. Using the *Xmx* parameter in the *h2o.init()* function, we can set aside the amount of RAM we want to use. I have 8GB of RAM on my machine; so I allocated 3GB to H2O.

```
library(h2o)
library(readr)
h2o.init(nthreads=-1)
localH2O = h2o.init(ip = "localhost", port = 54321,
startH2O = TRUE, Xmx = '3g')
cat("reading the training and testing data\n")
trainlocal_full <- read_csv("path\\train.csv")
Testlocal_full <- read_csv("path\\test.csv")
cat("loading into h2o")
train <- as.h2o(trainlocal)
test <- as.h2o(testlocal)
```

## Training a deep learning model

Once the data is loaded, the *h2o.deeplearning()* method with appropriate parameters can be used to invoke the deep learning engine. The H2O deep learning package has various methods to pre-process the data itself. However, if we understand and know something about our data, we can very well use R's native packages to pre-process it.

Shown below is sample code to invoke H2O's deep learning package. The model I settled on has the following attributes:

- Rectified linear units as the activation function
- An input drop-out ratio of 0.2
- A hidden drop-out ratio of 0.5
- Neuron architecture of 784-800-800-10 (784 inputs, two hidden layers of 800 neurons each, and 10 Softmax output neurons)
- 500 epochs
- ADADELTA adaptive learning rate

```
s <- proc.time()
set.seed(1105)
model <- h2o.deeplearning(x = 2:785,
 y = 1,
 data = train,
 activation = "RectifierWithDropout",
 input_dropout_ratio = 0.2,
 hidden_dropout_ratios = c(0.5, 0.5),
 balance_classes = TRUE,
 hidden = c(800, 800),
 epochs = 500)
e <- proc.time()
```

```
d <- e - s
d
model
```

Here is a brief overview of the parameters used.

- *X and Y*: List of the predictors and target variables, respectively
- *data*: H2O training frame data
- *activation*: Indicates which activation function to use
- *hidden*: Number of hidden layers and their size
- *l1*: L1 regularisation
- *train\_samples\_per\_iteration*: Number of training samples per iteration
- *classification\_stop*: Stopping criterion for classification error
- *epochs*: How many times the dataset should be iterated
- *balance\_classes*: If TRUE, balance the classes.

The model can be then applied to a new test dataset to validate its result using the *h2o.predict* package.

```
yhat <- h2o.predict(model, test)
ImageId <- as.numeric(seq(1, 28000))
names(ImageId)[1] <- "ImageId"
predictions <- cbind(as.data.frame(ImageId), as.data.
frame(yhat[, 1]))
names(predictions)[2] <- "Label"
write.table(as.matrix(predictions), file="training.csv", row.
names=FALSE, sep=",")
```

## Features that stand out for H2O

To summarise, the deep learning algorithm for IoT should have the following attributes:

- Be powerful enough to support distributed computing architectures to handle Big Data and do computations in parallel
- Be SMART enough
  - Be able to pre-process and auto-impute the data itself without any external intervention
  - Offer support for unsupervised feature learning
- Have the capability to prevent model overfitting
- Post process the data itself to give back results in the original form or unit of measure
- Cross-validate the results itself and decide if result optimisation is necessary

Almost all of these capabilities and features are demonstrated by H2O's deep learning package. This makes it an ideal predictive analytics engine and a suitable choice to implement deep learning for the Internet of Things. 

By: Srijan Agarwal

The author is a developer at WikiToLearn. He is an open source enthusiast and works mostly with JS and PHP. You can contact him at [srijanagarwal.cse@gmail.com](mailto:srijanagarwal.cse@gmail.com).

# Develop Real-time Applications with WebSockets and Socket.IO, Using Node.js



WebSockets is really useful for big applications, and your clients will definitely be impressed with the blazing speed it provides. Over time, it has improved considerably and got more stable. We might see a lot of scope for this technology very soon.



**R**eal-time applications are those for which there is an instant response to a user request. Examples include your phone ringing when another person is calling you, or you receiving a text message from others, or even your MP3 player (in a way) using a real-time mechanism when you rock it to switch songs, forward or backward.

How do real-time applications work? First, let's discuss real-time Web applications only, as real-time systems are very complex and hard to describe.

## Polling

Polling is the simplest real-time application mechanism. The mechanism operates as follows: after every regular interval, a request is sent to the server to check for updates in the form of a vote; the server then checks for any updates and sends an up-vote if updates are available and if not, it replies with a down-vote. At the user end, if the server casts an up-vote, the system further asks for the updated data which is then passed by the server. Once this process completes, again, requests are sent to the server at the regular intervals.

Ajax is preferred for polling as it is lightweight and can update parts of the page without refreshing the entire page. The problem is with the bandwidth, of which a large amount is used by both the server and the client. To reduce this, a method called *long-polling* has been introduced, whereby an HTTP request stays up until a new message is sent by the server. Gmail Chat is an example of this mechanism.

## WebSockets

Some applications need very little or almost no latency in client-server communication. This is only possible when there is persistent communication between the browser and the server, as it works in a bi-directional communication between the server and the client. Eventually, you can perform all the polling tasks by using WebSockets with minimum latency. Another great feature that WebSockets allows is browser-to-browser communication. This means that two client browsers can communicate with each other without any latency, while the server processes and stores data.

## Applications of WebSockets

WebSockets has become quite popular among Web developers these days, and can be used for the following applications:

- Chat application: Video and audio conferencing applications
- Multi-player online gaming
- Social feeds
- Simultaneous file sharing

There are endless applications that can be built with WebSockets, like stock market monitoring, sports updates, an online broadcast, etc. The question is how to use them? There are two incredibly great JavaScript libraries, namely, HTML5Rocks and Socket.IO. HTML5Rocks is natively available in modern browsers, while Socket.IO boasts of being the fastest among the competition. We personally prefer

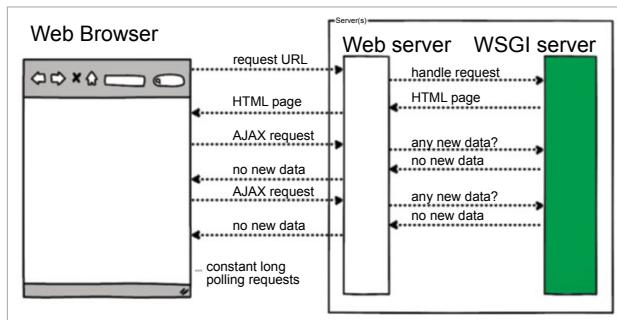


Figure 1: Long polling via AJAX

using Socket.IO, as it is really good and compatible with most of the mobile and desktop browsers used by people today. It is also based on Node.js and thus is very reliable for backend functionality too.

## An introduction to Socket.IO

As discussed earlier, Socket.IO is a JavaScript library used to build real-time applications. We personally prefer it as it is nearly platform-independent -- the applications tend to function across all the browsers without hiccups. Another reason is that Node.js has a great, lightweight JavaScript library to perform back-end functions.

Socket.IO boasts of being 'the fastest and most reliable real-time engine'. Installing it is really simple.

So let us start by installing Socket.IO in your project. At present, there is just one way to install it -- through npm (node package manager). For that, you have to open up your terminal and type the following command (you need to have Node.js installed on your machine):

```
$ cd /path/to/project_dir
$ npm install socket.io
```

## Getting started

Before we start on this, you are required to at least have some knowledge of the fundamentals of Node.js. Let us create a project named *test* for which we need the following:

- A computer/laptop with Windows, Linux or Mac installed with Node.js
- A text-editor like Notepad++, Emacs, Atom, Sublime, etc. Create a project folder named *test* on your desktop (or wherever you prefer).

As the Socket.IO application is made using Node.js, the first and most important rule is to create a *package.json* file in the *test* folder with the following code:

```
{
 "name": "test_application",
 "version": "0.0.1",
 "private": "true",
 "dependencies": {
 "socket.io": "1.4.8",
```

```
 "express": "4.14.0"
}
```

Now, open your terminal/cmd and use the following commands:

```
$ cd /path/to/project_dir
$ npm install
```

After the process completes, browse the project folder (*test*) and you will find a *node\_modules* directory. Open it and you will see the Socket.IO and Express directories there.

Now, make a new file named *test.js* and write the following code in it:

```
var express = require('express'),
 test = express(),
 server = require('http').createServer(test),
 io = require('socket.io').listen(server);
server.listen(3000);
test.get('/', function(req, res){
 res.sendFile(__dirname + '/welcome.html');
});

io.sockets.on('connection', function(socket){
 socket.on('sent', function(data){
 io.sockets.emit('receive', data);
 });
});
```

Here, the Express server is initialised into the application and Socket.IO is initialised with the same server so that it is embedded into the app for further usage. The server is then started on Port 3000 on your local machine and can be accessed by using the URL *localhost:3000*.

When someone tries to access the URL while the server is active, a *get* request is passed on to the server; the request is then processed under the label 'req' and the response to this is labelled 'res' in terms of variables. The application then executes the function, which asks it to deploy the Web page named 'welcome.html' which resides in the same directory as *test.js*. Let's now create a file named *welcome.html* and put the following code in it:

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>Hurray!</title>
 </head>
 <body>
 <div id="content">
 <form id="name_form">
```

```

Your name:
<input size="40" type="text" id="name" />
<input type="submit" />
</form>
</div>
<script src="https://code.jquery.com/jquery-3.1.0.min.js">
</script>
<script type="text/javascript" src="/socket.io/socket.io.js"></script>
<script>
jQuery(function($){
 var socket = io.connect();
 var $nameForm = $('#name_form');
 var $name = $('#name');

 $nameForm.submit(function(e){
 e.preventDefault();
 socket.emit('sent', $name.val());
 $name.val('');
 });

 socket.on('receive', function(data){
 var html = '<h1>Welcome ' + data + '</h1>
<i>This message was generated using socket.io</i>';
 $('#content').html(html);
 });
});
</script>
</body>
</html>

```

When done, go to the terminal and at the command prompt, type the following command:

```
$ node test
```

Open your browser and type *localhost:3000*.

If you see the result shown in Figure 2, congrats! You did everything exactly right. Here, we linked a *socket.io* file in the script section or mark up, which provided us the client-side functionalities of *socket.io*. Later, using our form, we ask for an input from our user and when he submits the form, we trigger the *emit* function of the socket, which sends the filled-out data to our Express server, which then processes our data and sends it back to the client using another *emit* function. Each *emit* function is uniquely named, like 'send' and 'receive'.

 **Note:** All this should be done without refreshing the page! To ensure that, we use jQuery's *preventDefault* function when the form is submitted.

Now, you are ready to move to the most important part of this article.

## Building a real-time application

We will now implement a LAN based chat engine, which you can use to communicate with all the people connected on the same network as you. But first, it's important to know how to access your apps on the local machine and access all the machines connected on the same network as your machine (your LAN).

Now, you can access your test app from any PC connected to your local network by just typing the IP address like this: "192.168.1.205:3000".

Just as in our previous test application, we need to create a *package.json* file and put the following code in it:

```
{
 "name": "test_chat_application",
 "version": "0.0.1",
 "private": "true",
 "dependencies": {
 "socket.io": "1.4.8",
 "express": "4.14.0"
 }
}
```

Let's run the script now by executing the following statement:

```
$ npm install
```

Next, let's create a file named *server.js*, which will work like *test.js* and use this code for your *server.js* file:

```

var express = require('express'),
app = express(),
server = require('http').createServer(app),
io = require('socket.io').listen(server),
nicknames = [];

server.listen(3000);

app.get('/', function(req, res){
 res.sendFile(__dirname + '/chat.html');
});

io.sockets.on('connection', function(socket){
 socket.on('new user', function(data, callback){
 if(nicknames.indexOf(data) != -1){
 callback(false);
 } else{
 socket.nickname = data;
 nicknames.push(socket.nickname);
 io.sockets.emit('usernames', nicknames);
 callback(true);
 updateNicknames();
 }
 });
});

```

```

function updateNicknames(){
 io.sockets.emit('usernames', nicknames);
}

socket.on('send message', function(data){
 io.sockets.emit('new message', {msg: data, nick: socket.nickname});
});

socket.on('disconnect', function(data){
 if(!socket.nickname) return;
 nicknames.splice(nicknames.indexOf(socket.nickname), 1);
 updateNicknames();
});

```

The code is divided into three sections:

1. To start up the server
2. For user management
3. For sending and receiving messages

We should now discuss our client-side HTML file named *chat.html*:

```

<!DOCTYPE html>
<html>
<head>
<title>local chat</title>
<style type="text/css">
body{
 font-weight: 400;
 font-family: sans-serif;
}
h1{
 font-weight: 300;
 color: green;
 text-align: center;
 font-size: 24px;
}
#contentWrap{
 display: none;
}
#chatWrap{
 padding: 10px;
 border: 2px solid black;
 width: 300px;
 box-sizing: border-box;
 background: lightyellow;
 float: left
}
#chat{
 overflow-y: scroll;
 height: 500px;
 border: 1px solid green;
}

width: 275px;
background: white;
padding: 0 10px;
box-sizing: border-box;
margin-bottom: 30px;
}
#chat ul{
list-style-type: none;
padding: 0;
display: inline-block;
}

}
#chat ul li{
padding: 4px;
box-sizing: border-box;
border: 1px solid blue;
background: lightblue;
margin: 1px 0;
font-size: 14px;
width: 250px;
}
#chat ul li b{
border-right: 2px solid #000;
margin: 0 5px 0 0;
}
#chatWrap form input{
height: 15px;
margin: 5px;
padding: 5px 5px;
width: 250px;
padding: 10px;
box-sizing: border-box;
}
#contentWrap #users{
margin-left: 320px;
width: 200px;
border: 1px solid black;
background: linear-gradient(120deg, #eee, #fff);
}

</style>
</head>
<body>
<h1>local chat</h1>
<div id="nickWrap">
<p>Enter a username:</p>

```



Figure 2: The result of the *test* application

```
sameers-MBP:~ sameer$ ifconfig |grep inet
 inet6 ::1 prefixlen 128
 inet 127.0.0.1 netmask 0xff000000
 inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
 inet6 fe80::7ed1:c3ff:fe74:14ba%en1 prefixlen 64 scopeid 0x6
 inet 192.168.1.205 netmask 0xffffffff broadcast 192.168.1.255
 inet6 fe80::48f8:b0ff:fe88:2c82%awd10 prefixlen 64 scopeid 0x8
sameers-MBP:~ sameer$
```

Figure 3: Fetching the local IP address

```
<form id="setNick">
<input size="35" type="text" id="nickname" />
<p id="error"></p>
<input type="submit" value="Enter chatroom" />
</form>
</div>
<div id="contentWrap">
<div id="chatWrap">
<div id="chat"><ul id="cht"></div>
<form id="send-message">
<input type="text" id="message" placeholder="your message" />
</form>
</div>
<div id="users">
</div>
</div>

<script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
<script type="text/javascript" src="/socket.io/socket.io.js"></script>
<script>
jQuery(function($){
var socket = io.connect();
var $nickForm = $('#setNick');
var $nickError = $('#error');
var $nickBox = $('#nickname');
var $users = $('#users');
var $messageForm = $('#send-message');
var $messageBox = $('#message');
var $chat = $('#cht');

$nickForm.submit(function(e){
e.preventDefault();
socket.emit('new user', $nickBox.val(), function(data){
if(data){
$('#nickWrap').hide();
$('#contentWrap').show();
} else{
$nickError.html('sorry, that username has been taken, try another one!');
}
})});
});
```



Figure 4: Local chat

```
});
$nickBox.val('');
});

$messageForm.submit(function(e){
e.preventDefault();
socket.emit('send message', $messageBox.val());
$messageBox.val('');
});

socket.on('usernames', function(data){
var html = '<h2>online users:</h2><hr />';
for(i=0; i < data.length; i++){
html += data[i] + '
';
}
$users.html(html);
});

socket.on('new message', function(data){
$chat.append("" + data.nick + " " + data.msg +
"
");
});
})
</script>
</body>
</html>
```

Now, test this on other computers by accessing this application on using your IP. [END](#) 

By: Prof. Prakash Patel and Sameer Manek

Prof. Prakash Patel is working as assistant professor in the Information Technology Department of the Gandhinagar Institute of Technology, which is affiliated to GTU. You can contact at [prakashpatelit@gmail.com](mailto:prakashpatelit@gmail.com)

Sameer Manek is a passionate programmer. He holds a diploma in computer engineering from Nirma University. Contact him at [maneksameer1@gmail.com](mailto:maneksameer1@gmail.com).

# regmap: Reducing the Redundancy in Linux Code

This article gives an overview of regmap—how useful it is in factoring out common code from the Linux sub-system and how to effectively use it to write drivers in Linux.

**L**inux is divided into many sub-systems in order to factor out common code in different parts and to simplify driver development, which helps in code maintenance.

Linux has sub-systems such as I2C and SPI, which are used to connect to devices that reside on these buses. Both these buses have the common function of reading and writing registers from the devices connected to them. So the code to read and write these registers, cache the value of the registers, etc, will have to be present in both these sub-systems.

This causes redundant code to be present in all sub-systems that have this register read and write functionality.

To avoid this and to factor out common code, as well as for easy driver maintenance and development, Linux developers introduced a new kernel API from version 3.1, which is called regmap. This infrastructure was previously present in the Linux ASoC (ALSA) sub-system, but has now been made available to entire Linux through the regmap API.

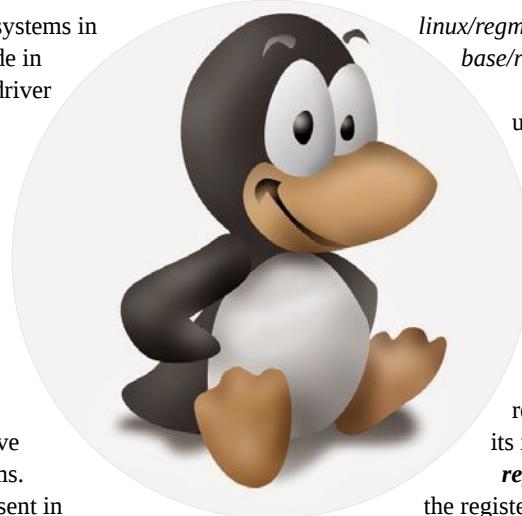
Earlier, if a driver was written for a device residing on the SPI bus, then the driver directly used the SPI bus read and write calls from the SPI sub-system to talk to the device. Now, it uses the regmap API to do so. The regmap sub-system takes care of calling the relevant calls of the SPI sub-system. So two devices—one residing on the I2C bus and one on the SPI bus—will have the same regmap read and write calls to talk to their respective devices on the bus.

This sub-system was first introduced in Linux 3.1 and later, the following different features were introduced:

- Support for SPMI, MMIO
- Spinlock and custom lock mechanism
- Cache support
- Endianness conversion
- Register range check
- IRQ support
- Read only and write only register
- Precious register and volatile register
- Register pages

## Implementing regmap

regmap in Linux provides APIs that are declared in *include/*



*linux/regmap.h* and implemented in *drivers/base/regmap/*.

Two important data structures to understand in Linux regmap are **struct regmap\_config** and **struct regmap**.

### **struct regmap\_config**

This is a per device configuration structure used by the regmap sub-system to talk to the device. It is defined by driver code, and contains all the information related to the registers of the device. Descriptions of its important fields are listed below.

**reg\_bits:** This is the number of bits in the registers of the device, e.g., in case of 1 byte registers it will be set to the value 8.

**val\_bits:** This is the number of bits in the value that will be set in the device register.

**writeable\_reg:** This is a user-defined function written in driver code which is called whenever a register is to be written. Whenever the driver calls the regmap sub-system to write to a register, this driver function is called; it will return ‘false’ if this register is not writeable and the write operation will return an error to the driver. This is a ‘per register’ write operation callback and is optional.

**wr\_table:** If the driver does not provide the *writeable\_reg* callback, then *wr\_table* is checked by regmap before doing the write operation. If the register address lies in the range provided by the *wr\_table*, then the write operation is performed. This is also optional, and the driver can omit its definition and can set it to NULL.

**readable\_reg:** This is a user defined function written in driver code, which is called whenever a register is to be read. Whenever the driver calls the regmap sub-system to read a register, this driver function is called to ensure the register is readable. The driver function will return ‘false’ if this register is not readable and the read operation will return an error to the driver. This is a ‘per register’ read operation callback and is optional.

**rd\_table:** If a driver does not provide a *readable\_reg* callback, then the *rd\_table* is checked by regmap before doing the read operation. If the register address lies in the range provided by *rd\_table*, then the read operation is performed. This is also optional, and the driver can omit its definition and can set it to NULL.

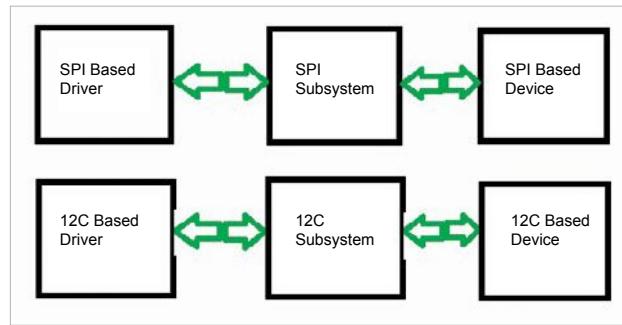


Figure 1: I2C and SPI driver before regmap

**volatile\_reg:** This is a user defined function written in driver code, which is called whenever a register is written or read through the cache. Whenever a driver reads or writes a register through the regmap cache, this function is called first, and if it returns ‘false’ only then is the cache method used; else, the registers are written or read directly, since the register is volatile and caching is not to be used. This is a ‘per register’ operation callback and is optional.

**volatile\_table:** If a driver does not provide a *volatile\_reg* callback, then the *volatile\_table* is checked by regmap to see if the register is volatile or not. If the register address lies in the range provided by the *volatile\_table* then the cache operation is not used. This is also optional, and the driver can omit its definition and can set it to NULL.

**lock:** This is a user defined callback written in driver code, which is called before starting any read or write operation. The function should take a lock and return it. This is an optional function—if it is not provided, regmap will provide its own locking mechanism.

**unlock:** This is user defined callback written in driver code for unlocking the lock, which is created by the lock routine. This is optional and, if it’s not provided, will be replaced by the regmap internal locking mechanism.

**lock\_arg:** This is the parameter that is passed to the *lock* and *unlock* callback routine.

**fast\_io:** regmap internally uses mutex to lock and unlock, if a custom lock and unlock mechanism is not provided. If the driver wants regmap to use the spinlock, then *fast\_io* should be set to ‘true’; else, regmap will use the mutex based lock.

**max\_register:** Whenever any read or write operation is to be performed, regmap checks whether the register address is less than *max\_register* first, and only if it is, is the operation performed. *max\_register* is ignored if it is set to 0.

**read\_flag\_mask:** Normally, in SPI or I2C, a write or read will have the highest bit set in the top byte to differentiate write and read operations. This mask is set in the higher byte of the register value.

**write\_flag\_mask:** This mask is also set in the higher byte of the register value.

These are the fields of the struct *regmap\_config*, and this configuration is passed to *regmap\_init*, which creates the struct *regmap* and returns what can be used in the read and

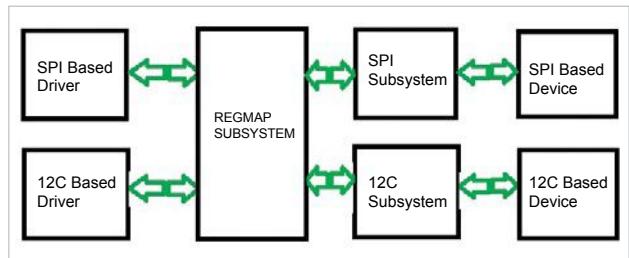


Figure 2: I2C and SPI driver after regmap

write operation.

*wr\_table*, *rd\_table* and *volatile\_table* are optional tables (used only if the respective callback is not provided) used by regmap for range-checking in case of write and read operations. These are implemented as *struct regmap\_access\_table*. The following are its fields.

**yes\_ranges:** These are address ranges that are considered as valid ranges.

**n\_yes\_ranges:** This is the number of entries in *yes\_ranges*.

**no\_ranges:** These are address ranges that are considered as invalid ranges.

**n\_no\_ranges:** This is the number of entries in *no\_ranges*.

## regmap APIs

regmap APIs are declared in *include/linux/regmap.h*. The following are the details of important APIs.

**Initialisation routines:** The following routine initialises regmap data structures based on the SPI configuration:

```
struct regmap * devm_regmap_init_spi(struct spi_device *spi,
const struct regmap_config);
```

The following routine initialises regmap data structures based on the I2C configuration:

```
struct regmap * devm_regmap_init_i2c(struct i2c_client *i2c,
const struct regmap_config);
```

In the regmap initialisation routine, the *regmap\_config* configuration is taken; then the regmap structure is allocated and the configuration is copied to it. The read/write functions of the respective bus are also copied in the regmap structure. For example, in the case of the SPI bus, the regmap read and write function pointer will point to the SPI read and write functions.

After regmap initialisation, the driver can talk to the device using the following routine:

```
int regmap_write(struct regmap *map, unsigned int reg,
unsigned int val);
int regmap_read(struct regmap *map, unsigned int reg,
unsigned int *val);
```

**regmap\_write:** This function is used to write data to the device. It takes in the regmap structure returned during

initialisation, registers the address and the value to be set. The following are the steps performed by the *regmap\_write* routine.

First, *regmap\_write* takes the lock, which will be spinlock if *fast\_io* in *regmap\_config* was set; else, it will be mutex.

Next, if *max\_register* is set in *regmap\_config*, then it will check if the register address passed is less than *max\_register*. If it is less than *max\_register*, then only the write operation is performed; else, *-EIO* (invalid I/O) is returned.

After that, if the *writeable\_reg* callback is set in *regmap\_config*, then that callback is called. If that callback returns ‘true’, then further operations are done; if it returns ‘false’, then an error *-EIO* is returned. This step is only performed if *writeable\_reg* is set.

If *writeable\_reg* is not set, but *wr\_table* is set, then there’s a check on whether the register address lies in *no\_ranges*, in which case an *-EIO* error is returned; else, it is checked whether it lies in the *yes\_ranges*. If it is not present there, then an *-EIO* error is returned and the operation is terminated. If it lies in the *yes\_ranges*, then further operations are performed. This step is only performed if *wr\_table* is set; else, it is skipped.

Whether caching is permitted is now checked. If it is permitted, then the register value is cached instead of writing directly to hardware, and the operation finishes at this step. If caching is not permitted, it goes to the next step. Caching will be discussed later.

After this hardware write routine is called to write the value in the hardware register, this function writes the *write\_flag\_mask* to the first byte of the value and the value is written to the device.

After completing the write operation, the lock that was taken before writing is released and the function returns.

**regmap\_read:** This function is used to read data from the device. It takes in regmap structure returned during initialisation, and registers the address and value pointer in which data is to be read. The following steps are performed during register reading.

First, the read function will take a lock before performing the read operation. This will be a spinlock if *fast\_io* is set in *regmap\_config*; else, regmap will use mutex.

Next, it will check whether the passed register address is less than *max\_register*; if it is not, then *-EIO* is returned. This step is only done if *max\_register* is set greater than zero.

Then, it will check if the *readable\_reg* callback is set. If it is, then that callback is called, and if this callback returns ‘false’ then the read operation is terminated returning an *-EIO* error. If this callback returns ‘true’ then further operations are performed.



**Note:** This step is only performed if *readable\_reg* is set.

What is checked next is whether the register address lies in the *no\_ranges* of the *rd\_table* in config. If it does, then an *-EIO* error is returned. If it doesn’t lie either in the *no\_ranges* or in the *yes\_ranges*, then too an *-EIO* error is returned.

Only if it lies in the *yes\_ranges* can further operations be performed. This step is only performed if the *rd\_table* is set.

Now, if caching is permitted, then the register value is read from the cache and the function returns the value being read. If caching is set to bypass, then the next step is performed.

After the above steps have been taken, the hardware read operation is called to read the register value, and the value of the variable which was passed is updated with the value returned.

The lock that was taken before starting this operation is now released and the function returns.

## Writing a regmap based driver

Let us try to write a driver based on the regmap framework.

Let us assume that there is a device X connected on an SPI bus, which has the following properties:

- 8-bit register address
- 8-bit register values
- 0x80 as write mask
- It’s a fast I/O device so the spinlock should be used
- Valid address range:
  - 0x20 to 0x4F
  - 0x60 to 0x7F

The driver can be written as follows:

```
//include other include files
#include <linux/regmap.h>

static struct custom_drv_private_struct
{
 //other fields relevant to device
 struct regmap *map;
};

static const struct regmap_range wr_rd_range[] =
{
 {
 .range_min = 0x20,
 .range_max = 0x4F,
 },
 {
 .range_min = 0x60,
 .range_max = 0x7F
 },
};

struct regmap_access_table drv_wr_table =
{
 .yes_ranges = wr_rd_range,
 .n_yes_ranges = ARRAY_SIZE(wr_rd_range),
};

struct regmap_access_table drv_rd_table =
{
 .yes_ranges = wr_rd_range,
 .n_yes_ranges = ARRAY_SIZE(wr_rd_range),
```

```

};

static bool writeable_reg(struct device *dev, unsigned int
reg)
{
 if(reg >= 0x20 && reg <= 0x4F)
 return true;

 if(reg >= 0x60 && reg <= 0x7F)
 return true;

 return false;
}

static bool readable_reg(struct device *dev, unsigned int
reg)
{
 if(reg >= 0x20 && reg <= 0x4F)
 return true;

 if(reg >= 0x60 && reg <= 0x7F)
 return true;

 return false;
}
static int custom_drv_probe(struct spi_device *dev)
{
 struct regmap_config config;
 struct custom_drv_private_struct *priv;
 unsigned int data;
 //configure the regmap configuration
 memset(&config, 0, sizeof(config));
 config.reg_bits = 8;
 config.val_bits = 8;
 config.write_flag_mask = 0x80;
 config.max_register = 0x80;
 config.fast_io = true;
 config.writeable_reg = drv_writeable_reg;
 config.readable_reg = drv_readable_reg;
 //only set below two things if
 //writeable_reg
 //and readable_reg is not set
 //config.wr_table = drv_wr_table;
 //config.rd_table = drv_rd_table;

 //allocate the private data structures as
 //priv = devm_kzalloc
 //Init the regmap spi configuration
 priv->map = devm_regmap_init_spi(dev,
 &config);
 //devm_regmap_init_i2c in case of i2c bus
 //following operation will remain same in
 //case of both i2c and spi or other bus
 //read from the device, data variable will //contain device
}

```

```

data
regmap_read(priv->map, 0x23, &data);
data = 0x24;
//write to the device
regmap_write(priv->map, 0x23, data);
if(regmap_read(priv->map, 0x85, &data)
< 0)
{
 //error since address is out of range
}
return 0;
}

```

You can see in the above example how redundant code in the drivers based on different bus sub-systems becomes similar code, which makes driver writing and maintenance easier. Cache support is also available in the current regmap sub-system.

Caching avoids directly performing operations on the device. Instead, it caches the value transferring between the device and driver, and uses it as future reference. Initially, caches used just flat arrays, which were not good for 32-bit addresses. Later, this issue was solved with better cache types:

- *rbtree* stores blocks of contiguous registers in a red/black tree
  - *Compressed* stores blocks of compressed data
- Both rely on the existing kernel library:

```
enum regcache_type cache_type;
```

Support for tracepoint is available in regmap. For more information, see *debugfs/trace/events/regmap*:

```
regmap_reg_write 0-001b reg=3b val=1a
regmap_reg_read 0-001b reg=1d val=1d
```

You can also define *LOG\_DEVICE* for early init log 

## References

- [1] DS1343 driver file: <https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/drivers/rtc/rtc-ds1343.c>
- [2] [http://elinux.org/images/a/a3/Regmap\\_-The\\_Power\\_of\\_Subsystems\\_and\\_Abstractions.pdf](http://elinux.org/images/a/a3/Regmap_-The_Power_of_Subsystems_and_Abstractions.pdf)
- [3] <http://free-electrons.com/pub/conferences/2015/elce/belloni-mfd-regmap-syscon/belloni-mfd-regmap-syscon.pdf>
- [4] <http://www.sirena.org.uk/2011/09/30/regmap-a-register-map-abstraction-for-the-linux-kernel/>

## By: Raghavendra Chandra Ganiga

The author is an embedded firmware development engineer at K2 Technosoft Pvt Ltd, Pune. His interests are microcontrollers, networking firmware, RTOS development and Linux device drivers. You can contact him at [ravi23ganiga@gmail.com](mailto:ravi23ganiga@gmail.com).

# A Few Tips on Vi/Vim Editor for Linux Newbies

Inviting all Linux newbies, and even veterans, to try out these tips on using Vi/Vim editor efficiently and productively.



**E**arlier, most Linux users cut their teeth on Linux with the Vi editor. Those comfortable with using the GUI for Linux may scoff at using Vim for editing, but it does have a few advantages, particularly when working in multi-user systems and also when working on servers. This is because of the Vi editor's efficiency and its minimal use of memory. It is powerful, highly configurable and always available on any system.

I have compiled a few Vi/Vim editor tips, which will make it easier for readers to use it and become more productive.

## Abbreviations

Abbreviations can be set, which Vi will expand into full text whenever they are typed in edit mode. An abbreviation can be defined in the command mode by giving the following *EX* command:

```
:ab abbr full text
```

...where *abbr* is an abbreviation for the specified phrase. The expansion is done only when the abbreviation is typed as a word. Avoid repeating the abbreviation in the

defined phrase.

The abbreviation can be disabled by typing the following command:

```
:unab abbr
```

## Encryption in Vi

To encrypt any file in the Vi editor, type the following in command mode and press the *Enter* key:

```
:X
```

You will be asked to enter a password twice, which will be used to encrypt the file. The file is now saved in an encrypted format. When opening that file, the password needs to be re-entered. Of course, if you have encrypted a file containing a programming language source code, the file cannot be compiled.

To decrypt the encrypted file, open the file using Vi and type *:X*. Next, hit the *Enter* key twice; then save and exit the Vi editor by typing *:wq*. The file will be saved in the decrypted format.

## The `.vimrc` file

The `EX` commands (those that you key in after typing : in the Vi editor), which you want to execute whenever you start the Vi editor, can be saved in the `.vimrc` file in your home directory. If this file is not already there, you can create it. The `.vimrc` file can be modified like any other text file. Examples of some settings are given in Table 1.

Table 1

Command	Short form	Action
set number	<code>se nu</code>	Displays line numbers
set nonumber	<code>Se nonu</code>	Removes line numbers
set autoindent	<code>se ai</code>	Automatic indentation in insert mode
Set noautoindent	<code>Se noai</code>	Removes the automatic indentation feature in insert mode
<code>se tabstop=3</code>	<code>se ts=3</code>	Sets the number of spaces by which the tab indents during editing
<code>syntax on</code>		Turns on highlighting of syntax
<code>syntax off</code>		Turns off highlighting of syntax
<code>set shiftwidth=4</code>	<code>set sw=4</code>	The size of the indent, measured in spaces

## Inserting the contents of an existing file

Often, we need to insert the contents of an existing file into the current file. The Vi editor provides an easy way to do this. Take the cursor to the position at which you want the insertion and issue the following command:

`:r filename`

This inserts the contents of the file from the current cursor position onwards.

Adding a number before `r` inserts the text file below the line with the particular line number:

`:40r filename`

## Running an external command within Vi

You can execute any command within the Vi editor by typing the following command within the command mode of Vi:

`:!commandname`

## Insert the output of a Linux command

You can insert the output of a command by giving the following command:

`:r! :commandname`

The command is executed and the output is inserted from the position where the cursor is located.

## Opening Vi directly at a particular line

You can open the Vim editor with the cursor placed directly at the line you want, by giving the following command:

`$vim filename +n`

The number 'n' in the command is the line number.

## Opening Vi at the end of the file

Adding the + sign and a space before the file name, opens the file and places the cursor at the last line. This is useful if you have to work on large files.

`$vim + filename`

## Writing the buffer to a new file

`:w` can be used to save the complete buffer, that is, the file being edited currently, under a new file name. This can be used when you have made a lot of changes in the file, but then realise that you do not want to overwrite the original file. To save the contents of the buffer, use the following command:

`:w newname`

You can now quit the original file by typing `:q`.

## Indenting the source code

In case you have typed dozens of lines of code without bothering about indentation, the quickest way to indent the code is to do the following:

Press the `Esc` key to go to command mode and then type the following:

`gg=G`

...where `gg` indicates the beginning of the file, = is for indenting and `G` indicates the end of the file.

## Repeating the last change

We can use the . (period) key to repeat the last change. This helps to reduce the typing required when carrying out repetitive tasks.

## Undoing and redoing

Type `u` to undo the last change.



```
// trial for indentation
#include <stdio.h>
main()
{
 int i,j, size;
 for (i=1; i <= 10; i++)
 {
 printf ("\n");
 for (j=1; j <= 10; j++)
 {
 printf (" %d ", j);
 }
 printf ("\n");
 }
 printf ("\n");
}
```

Figure 1: A code snippet without indent

```

1 /* trial for indentation
2
3 #include <stdio.h>
4
5 main()
6 {
7 int i,j, size;
8
9 for (i=1; i <= 10; i++)
10 {
11 printf ("\n");
12 for (j=1; j <= 10; j++)
13 {
14 printf ("%d ", j);
15 }
16 printf ("\n");
17 }
18 printf ("\n");
19 }

```

Figure 2: A code snippet with indent

Type *Ctrl-R* to repeat a change that has been undone.

## Jumping to matching braces

We can jump to the matching parenthesis (the curly brace or square bracket) by pressing the *%* key. This is very useful when editing source code.

## Preventing auto-indent

When text is pasted in Vim from an already indented source file, Vim will further apply its auto-indenting feature (if it

is enabled) to the pasted text, which will have a cascading effect on the text. To prevent auto-indenting, use the following command:

`:set pastetoggle=<F2>`

You can then press *<F2>* in the insert mode when you are ready to paste. After pasting, you can press *<F2>* again to go to the auto-indent mode.

## Vimtutor

You can type *vimtutor* at the prompt and go through the tutorial on Vim. It will help you to learn more about Vim, quickly. 

**By: S. Sathyaranarayanan**

The author works as the information scientist in the Sri Sathy Sai Institute of Higher Learning, Brindavan Campus, Bengaluru, and also heads the computer centre of the campus. He has more than 25 years of experience in systems administration and in teaching IT courses, and is an enthusiastic promoter of FOSS. He can be reached at [sathyaranayanan\\_s@yahoo.com](mailto:sathyaranayanan_s@yahoo.com) / [ssathyaranayanan@sssihl.edu.in](mailto:ssathyaranayanan@sssihl.edu.in).

# support@efy.in

**Do you have a query,  
suggestion or a complaint?**

**You can e-mail it to  
[support@efy.in](mailto:support@efy.in)  
and we will take care  
of the rest.**





with **8000+**  
registrations, we lived  
up to our tag of

# Asia's #1 Open Source Conference



# Thank You

**VISITORS**, for making the event a huge success

**SPEAKERS**, for contributing your valuable thoughts

**PARTNERS**, for your support



*13th Edition*

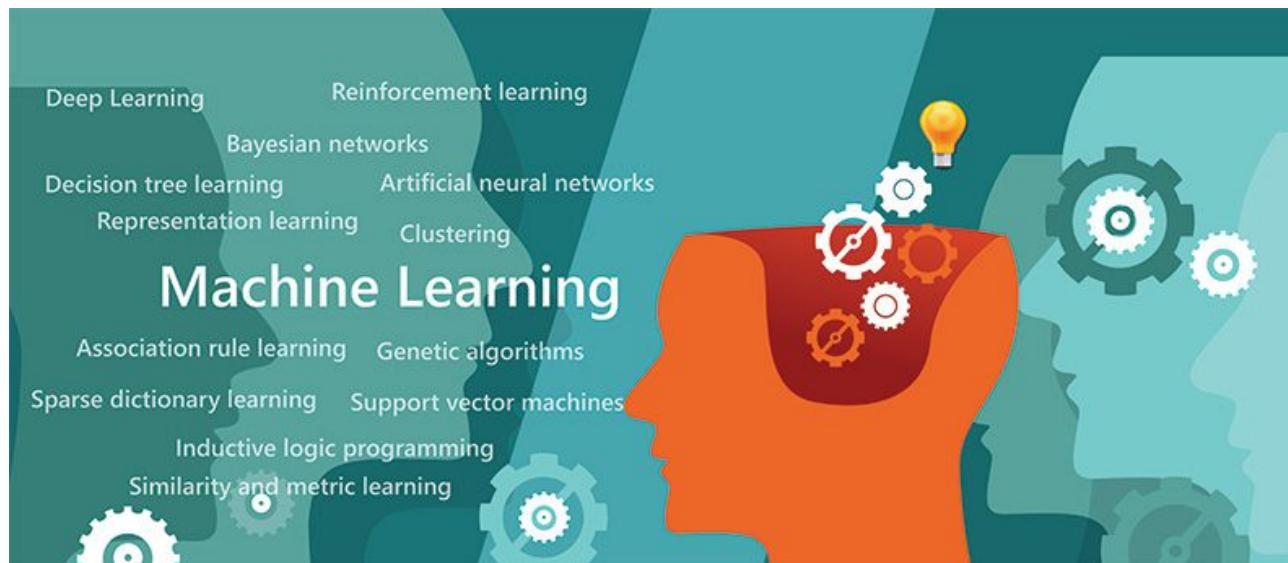
**OPEN**  
**SOURCE INDIA**

**21-22**  
**October**  
**2016**

NIMHANS Convention Center

**BENGALURU**

# Machine Learning Basics for Newbies



Machine learning focuses on the development of computer programs that can teach themselves to grow and adapt when exposed to new data. It is increasingly impacting our lives nowadays, as machines play an important role in banking and financial services, healthcare, retail, publishing, and in the social media, robot locomotion and gaming domains.

When we start the journey of life as new born babies, we inherit the characteristics of our parents. We don't know what to do and when to do what. As we grow up, our parents and elders teach us how to walk, talk and take various decisions in our lives and, as time passes, we gain experience and knowledge. Finally, we start taking our own decisions based on our learning and experience. Similarly, when we write any code to make a system do any work, the system only does what we ask it to do—it cannot think or take any extra decisions on its own nor perform actions on that basis. Yet, machine learning actually teaches the system to learn and take decisions when exposed to a new set of data on the basis of the experience it gains while performing different actions. Nowadays, it is an emerging technology that is widely being implemented across all types of industries. Google's self-driving cars, flying drones, anomaly detection or Big Data processing are among the recent examples of this technology being used.

Machine learning is one type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. It uses pattern recognition and computational learning theory to study and develop algorithms (which can learn from the sets of available data), on the basis of which it makes decisions. These algorithms work by building a model (such as the Predictive Model or Neural Network Model) from sample inputs in order to make data-driven decisions. These models help in developing

Decision Trees, using which, the system makes its decision.

Machine learning makes use of mathematical optimisation to deliver different theories, methods and application domains for a specific field. It uses the data mining technique to perform exploratory data analysis over a set of data in order to make predictions. This is basically referred to as unsupervised learning. Machine learning helps data scientists, engineers, researchers and analysts to take a reliable decision by uncovering the hidden insights acquired through the analysis of historical trends in the data.

## Types of machine learning

The tasks performed using machine learning are classified broadly into three categories, based on the nature of the learning signal available to a learning system (the system which helps to make decisions).

1. *Supervised learning:* This is a type of machine learning in which the system is presented with a set of labelled training data (inputs and their corresponding set of outputs). Now, it is the task of supervised machine learning to predict a new set of outputs for a given new set of inputs by learning or finding out a general rule or pattern that maps the given set of inputs to their corresponding outputs. The pattern or rule that helps in predicting output is generally denoted by a specific function. Supervised learning is further classified as regression and classification problems, on the basis of the methodology that is implemented to find a specific pattern.

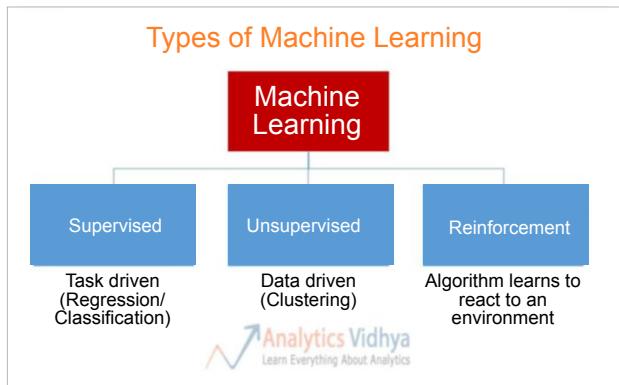


Figure 1: Classification of machine learning techniques  
(image credits: [www.analyticsvidhya.com](http://www.analyticsvidhya.com))

- Unsupervised learning:** This is a machine learning technique which is implemented when there is only a set of inputs available with the system, with no corresponding outputs. Now it's left to the system to learn and identify the pattern or rule governing the available inputs by using unsupervised learning and, further, that hypothesis or rule is used to find the output for the given set of inputs. There can be many possible hypotheses, but the optimal one out of all of them is considered for finding the output. Again, the unsupervised learning technique is further classified as K Means and Hierarchical Clustering problems, on the basis of the different techniques used to find the final hypothesis.
- Reinforcement learning:** Here the system is given two different sets of input data and it needs to implement the Reinforcement Machine Learning technique in order to learn and identify the general pattern or hypothesis in one of the given set of inputs. There can be more than one hypothesis derived but, finally, the optimal one out of all those available is used by the system to derive the output for the other set of inputs. This is like learning the rules of a game by playing against an opponent.

## Implementation of machine learning in real life scenarios

Let's now look at implementing machine learning in real life scenarios. We need to check how we can teach machines to make decisions and do our work just as we would do it, by applying our own sense or logic.

Actually, in the course of teaching machines, every stage of the process helps to build a better version of the machine. There are five basic steps that need to be followed prior to letting a machine perform any unsupervised task.

- Collecting data:** This is one of the first and foremost steps in implementing any type of machine learning technique. The data plays quite a significant role in machine learning, whether it is in the form of raw data from MS Excel, Access or even text files. This step lays the foundation of future learning. We must be aware of the fact that the

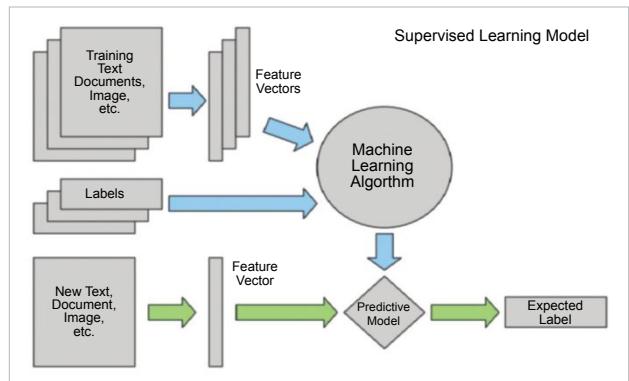


Figure 2: Supervised learning model (image credits: Google images)

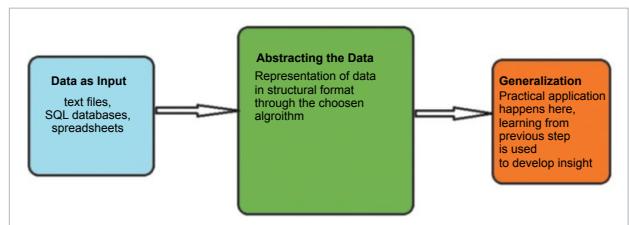


Figure 3: Steps to implement machine learning

- better the variety, volume and density of relevant data, the better will be the learning prospects for the machine.
- Preparing the data:** Once the data is collected, we need to check the quality of what will be fed as training data to the system. We need to spend time in order to determine the quality of data and, accordingly, take steps to fix issues such as treatment of outliers and missing data. Exploratory analysis is one such methodology used to study the differences of the data in detail, thereby strengthening the 'nutritional content' of the data.
  - Training a model:** This step involves selecting the appropriate algorithm and representing data in the form of a model. The final purified data is split into two parts – *training* and *test* (the proportion of the data depends on the prerequisite requirements). The first part (training data) is used to develop the model, whereas the second part (test data) is used as the reference.
  - Evaluating the model:** This step involves the evaluation of the machine learning model we chose to implement. The second part of the data (test data) is used to test the accuracy of the learning model. This step actually determines how precise the algorithm selected is, based on the outcome. There is also a better test to check the accuracy of the model, which sees how the model performs on data that has not been used at all while building it.
  - Improving the performance:** This step may involve choosing a different model altogether or even introducing more variables to improve the efficiency of the learning model. If the model is changed, then it again needs to be evaluated and its performance checked, which is why a lot of time needs to be spent in collecting and preparing data.

## Open source libraries, frameworks and tools to implement machine learning

In order to implement machine learning on a system for any scenario, there are enough open source tools, software or frameworks available for us to choose from, based on our preference for a specific language or environment. Let's have a look at some of them.

**Shogun:** Shogun is one of the oldest and most venerable of the different machine learning libraries available in the market. It was first developed in 1999 using C++, but now it's not limited to working in C++ only; rather, it can be used transparently in many languages and environments such as Java, C#, Python, Ruby, R, Octave, Lua and MATLAB. It is very easy to use, and quite fast at compilation and execution. As it can be used in a wide range of languages, people feel comfortable with it.

**Weka:** Weka was developed at the University of Waikato in New Zealand. It collects a set of Java machine learning algorithms which are engineered specifically for data mining. This GNU GPLv3-licensed collection possesses a package system, which can be used to extend its functionality. It has both official and unofficial packages available. Weka comes with a book that explains both the software and the techniques used in it. While Weka is not aimed specifically for Hadoop users, it can be used with Hadoop as well, because of the set of wrappers that have been produced for the most recent versions of it. It doesn't support Spark, but Clojure users can also use Weka.

**CUDA-Convnet:** CUDA-Convnet is a machine learning library especially used for neural-network applications. It is written in C++ in order to exploit Nvidia's CUDA GPU processing technology. It can even be used by those who prefer Python rather than C++. The resulting neural nets obtained as output from this library can be saved as Python pickled objects and, hence, can be accessed from Python.

We should note that the original version of the project is no longer being developed, but has been reworked into a successor named CUDA-Convnet2. It supports multiple GPUs and even Kepler-generation GPUs.

**H2O:** H2O is an open source machine learning framework developed by Oxdata. H2O's algorithms are basically geared for business processes, like fraud or trend predictions. H2O can easily interact in a standalone fashion with different HDFS stores. It can be in MapReduce, on top of YARN, or directly in an Amazon EC2 instance as well. Hadoop Mavens can use Java for interaction with H2O, but this framework also provides bindings for R, Python and Scala. It enables cross-interaction with all the libraries that are available on those platforms.

## The applications of machine learning in the current tech era

Guess we are all curious and interested to know the applications of machine learning in the current scenario

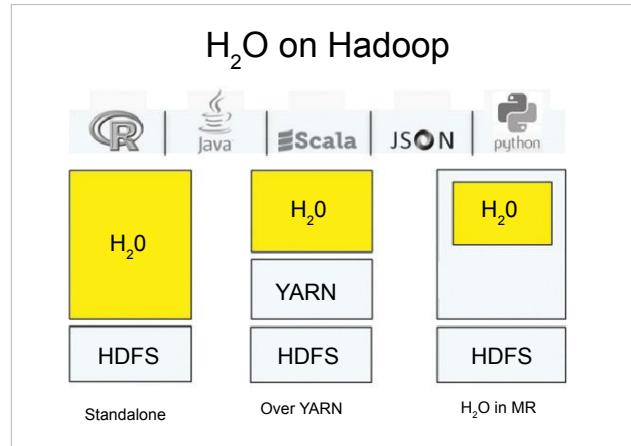


Figure 4: H2O on Hadoop (image credits: [www.infoworld.com](http://www.infoworld.com))

when the world is on the path to becoming smarter through the automation of all possible manual tasks. Google and Facebook use machine learning to push their respective advertisements to the relevant users. Here are a few applications that we should know of.

- Banking and financial services:** Machine learning is widely used to predict the customers who are likely to be defaulters in paying credit card bills or in repaying loans. This is of utmost importance as machine learning helps banks to identify the customers who can be given credit cards and loans.
- Healthcare:** It is widely used to diagnose various deadly diseases (like cancer) on the basis of patients' symptoms, and tallying them with the past data available for similar kinds of patients.
- Retail:** Machine learning is used to identify products that sell fast and those that don't. It helps retailers to decide on the kind of products to introduce or remove from their stock. Also, machine learning algorithms can be very effective in finding two or more products that will sell together. This is basically done to encourage customer loyalty initiatives which, in turn, help different retailers to develop and maintain loyal customers. Walmart, Amazon, Big Bazaar and all other such retail units extensively make use of machine learning.
- Publishing and social media:** There are different publishing firms like LexisNexis and Tata McGraw Hill, which make use of machine learning to run the queries and fetch the documents required by their users online, based on their preferences and requirements. Google and Facebook also use these techniques to rank their search outputs and news feeds. Facebook also provides a list of possible friends under its 'Friend suggestions' using this.
- Robot locomotion:** Robot locomotion is basically a collective term used for the different methods that robots use to transport themselves from one place to other. A major challenge in this field lies in developing capabilities for different robots to autonomously decide

about how, when and where to move. Machine learning helps them do this quite easily. Apart from this, there are various decisions that robots need to take instantaneously while they perform any activity, which is possible using different machine learning techniques.

6. *Game playing:* A strategy game is one in which the player's autonomous decision-making skills are quite significant in determining the final outcome. Almost all the strategy games require internal 'decision tree style' of thinking, and typically require very high situational awareness. Machine learning meets all these requirements and, hence, is widely used in gaming.

## Advantages of machine learning

1. Machine learning techniques help the system to take decisions on the basis of training data in dynamic or uncertain situations as well.
2. It can handle multi-dimensional, multi-variety data, and can even extract implicit relationships within large data sets in a dynamic, complex and chaotic environment.
3. It allows reduction of the time cycle and improves resource utilisation. It also provides different tools for continuous quality improvement in any large or complex process.
4. Another advantage of machine learning techniques is the increased usability of various applications of algorithms due to source programs like Rapidminer. Machine learning allows easy application and the comfortable adjustment of parameters to improve the classification performance.

## Challenges of machine learning

1. A very common challenge is the acquisition of relevant data.
2. Once the available data is secured, it often has to be pre-processed depending on the requirements of the specific algorithm used, which has a critical impact on the final results.
3. Sometimes, the interpretation of results also becomes a challenge, as these need to be interpreted according to the algorithm chosen.

Different machine learning techniques can be implemented in order to let the system decide on what action it needs to take and when it needs to be taken. Machine learning can really give an edge to automation, and has already helped in making the world a lot smarter. But machines haven't stopped learning, yet, and the next level of this technology is being worked on as I write this! 

## References

- [1] <http://www.wikipedia.org/>
- [2] <http://www.infoworld.com>
- [3] <https://www.analyticsvidhya.com>

## By: Vivek Ratan

The author currently works as an automation test engineer at Infosys, Pune. He can be reached at [ratanvivek14@gmail.com](mailto:ratanvivek14@gmail.com)

## OSFY Magazine Attractions During 2016-17

MONTH	THEME
March 2016	Open Source Databases
April 2016	Backup and Data Storage
May 2016	Web Development
June 2016	Open Source Firewall and Network Security
July 2016	Mobile App Development
August 2016	Network Monitoring
September 2016	Open Source Programming Languages
October 2016	Cloud Special
November 2016	Open Source on Windows
December 2016	Machine Learning
January 2017	Virtualisation (containers)
February 2017	Top 10 Special



## **Making man pages useful**

If you are looking for some help on a particular subject or command, man pages are a good place to start. You normally access a man page with the command *man <command>*. But you can also search the man pages' descriptions for a particular keyword. As an example, to search for man pages that discuss logins, use the following command:

```
#man -k login
```

When you access a man page, you can also use the forward *slash* key to search for a particular word within the man page itself. Simply press *'/'* on your keyboard and then type in the search term.

—*Pallavi Rawat, pallavifirst@rediffmail.com*

## **Mounting a Windows partition on Linux**

When using Linux as the dual boot option with Windows 8/8.1/10, users often get an error message while mounting their NTFS partitions, such as: 'The NTFS partition is in an unsafe state. Please resume and shutdown Windows fully', along with other technical details of the error.

This happens due to the FastBoot feature of Windows 8/8.1/10, which keeps NTFS partitions in hibernated mode when Windows is shut down.

Once the hibernate flag is set to 'On' for the partition, Linux cannot mount that partition and considers it is being used in Windows.

This can be avoided by turning the FastBoot feature off. This may not cause any trouble, apart from users noticing an increased shutdown time for the Windows operating system. But they won't mind this difference if they're not frequent Windows users.

In Windows 8/8.1/10:

1. Open the control panel and from the small icons, view and click on *Power Options*.
2. Click on *Choose what the power buttons do*.
3. Click on *Change settings that are currently unavailable*.
4. Uncheck *Turn on fast startup* (recommended).

Completely restart Windows, log in to Linux and try to mount the NTFS partitions.

—*Vinayak Kulkarni, kulkarnivd1989@gmail.com*

## **How to manage files in the Linux terminal, using commands**

To use the Linux terminal like a pro, you'll need to know the basics of managing files and navigating directories. True to the UNIX philosophy, each command does one thing and does it well. Midnight Commander, a full-featured file manager for the Linux terminal, acts as a powerful front-end to all these commands.

Most regular users know often-used commands like how to create, modify or delete. But the most important command to know about is how to view the full file manager.

```
mc - A Full File Manager
```

Midnight Commander is one of many full-featured file managers that you can use from the Linux terminal. It isn't installed by default on most distributions; so here's the command you'll need to install it on Ubuntu:

```
sudo apt-get install mc
```

After installation, just run the *MC* command. Use the arrow keys to select files and the tab key to switch between panes. Press *Alt-1* to see the *Help* screen or *Alt-2* for the menu.

You can also use the mouse in Midnight Commander if your terminal environment has mouse support.

—*Gangappan Venkatesan, vgangacse@gmail.com*

## **Removing all spaces or unwanted characters from the file name**

Often, there are multiple spaces or unwanted characters in the file name, and it becomes annoying to access these files from the terminal. We can remove spaces from file names manually, but it becomes a tedious task when there are a large number of files involved. Fortunately,

GNU/Linux has a command that solves this problem. This is the ‘rename’ command, which renames the file names supplied according to the rule specified as the first argument. Let us understand this with a simple example.

Shown below are the files whose names contain spaces:

```
[bash]$ ls -1
song 1.mp3
song 2.mp3
song 3.mp3
song 4.mp3
```

Now, let us remove these spaces using the ‘rename’ command as follows:

```
[bash]$ rename 's/ //g' *.mp3
```

It’s that simple; we are done. Now, let us verify the results:

```
[bash]$ ls -1
song1.mp3
song2.mp3
song3.mp3
song4.mp3
```

Indeed, it has worked. In the above example, the ‘rename’ command specified the rule to replace the space with nothing, i.e., it just removes the space. In the above command, the ‘g’ option implies ‘global’, i.e., ‘remove all spaces’. If you want to replace spaces with any other character—for instance, an underscore—then use the following rule with the ‘rename’ command:

```
[bash]$ rename 's/ /-/g' *
```

```
[bash]$ ls -1
song-1.mp3
song-2.mp3
song-3.mp3
```

—Narendra Kangralkar,  
narendrakangralkar@gmail.com

## Use Tree for visualising the directory tree structure

The *Tree* command is useful for visualising the directories and files residing in the current directory. To see what this command does, we need to install it first. Run the following command if you are on an Ubuntu based system:

```
sudo apt-get install tree
```

In Fedora-like systems, use the following command:

```
yum install tree
```

After installation, just type ‘tree’ and hit *Enter*. It will display the current directory tree structure with the file name in the respective directory.

If you just want to explore directories inside the current directory, type the following command:

```
tree -d
```

This will display the tree structure only with the listed directories. Other *tree* commands, and what they do, are listed below.

*tree -a*: Displays all files and directories

*tree -h*: Displays files and directories with their sizes

*tree -f*: Will display the tree structure with the full path to the file and directories

—Rajatkumar Zala, rajatkumarzala@yahoo.com



## How to find out how long the process has been running

Here is a simple tip to get the PID of your Apache process. Type:

```
[root@linuxraja ~]# ps -ef | grep httpd | grep root | grep -v grep
```

```
root 31096 1 0 2015 ? 00:00:52 /usr/sbin/
httpd
```

Now, we will check how long the process has been running:

```
ps -eo pid,etime | grep PID
```

```
[root@linuxraja ~]# ps -eo pid,etime | grep 31096
31096 152-22:29:16
```

The above output shows that the Apache process has been running for 152 days, 22 hours, 29 minutes and 16 seconds

—Natraj Solai, linuxraja@gmail.com



## Share Your Linux Recipes!

The joy of using Linux is in finding ways to get around problems—take them head on, defeat them! We invite you to share your tips and tricks with us for publication in *OSFY* so that they can reach a wider audience. Your tips could be related to administration, programming, troubleshooting or general tweaking. Submit them at [www.opensourceforu.com](http://www.opensourceforu.com). The sender of each published tip will get a T-shirt.

# “We come with ample expertise in MySQL”

Database management has become easy with the use of open source. There are plenty of solutions available in the market, which are backed by the community. **Jagmeet Singh** of **OSFY** talked with **Pradeep Chandru, founder and CEO of Mafiree**, to know about the latest developments in the world of database management. Chandru is a certified MySQL DBA and MySQL Cluster certified professional, and his team has over eight years of experience in database management. Here are edited excerpts from the interaction.



» **Pradeep Chandru**, founder and CEO of Mafiree

**Q** How are you exploring the capabilities of open source to support enterprises in India?

We come with ample expertise in MySQL. This has helped us fulfil the migration requirements of a large number of enterprise customers in India, with ease. We have shown our customers, through benchmarking, the improved performance a well-tuned open source solution can give them. We also leverage these use cases when we pitch for new enterprise customers.

**Q** Do you feel that open source has moved beyond just giving those starting a business a kickstart—that it has become a big movement, which is growing in the market?

Yes. In the current scenario, almost all types of industries have started exploring open source technologies to meet different needs. This includes even government bodies and

PSUs. This is mainly due to the wide variety of options available in open source technologies, and besides, the requirements of today's industries have also expanded to a great extent. The demand-supply balance in today's world cannot be met only with proprietary technologies. Open source plays a critical role here.

**Q** Why did you choose MySQL to kickstart your database management operations?

Before becoming one of the founders of Mafiree, I had ventured into the MySQL world for more than half a decade. Hence, there was a natural affinity towards that technology. Being an open source platform, MySQL has scaled up to a level that no other contemporary open source RDBMS solution has been able to. Therefore, we were able to predict the enormous potential that MySQL possessed and wanted to help other open source enthusiasts in harvesting its benefits. We started by cracking a big deal with Sun Microsystems involving a migration from Oracle to MySQL. This was hugely successful for us and helped us start on this journey on a positive note.

**Q** Is it easy to opt for an open source solution like MySQL or PostgreSQL over Oracle and SAP to manage large data logs?

We were able to successfully migrate quite huge systems from Oracle to MySQL/Postgres. We have migrated systems that had more than 10TB of data and 4,500 concurrent users from an Oracle environment to MySQL, without degrading the performance. This is because we took advantage of many architectural features that were available in MySQL, which proved to be better than Oracle.

**Q** What are the challenges in migrating a database from a proprietary solution such as Oracle and SAP to an open source one like MySQL?

In old legacy systems, we have seen people use packages which may not have a direct equivalent in MySQL and we may have to use a workaround to manage this migration. But nowadays, it has become easier to migrate applications that use inline queries compared to a legacy application that uses stored processes, in which case we might need changes to be done in stored procedures.

**Q** How is open source helping database management nowadays?

Open source offers a lot of flexibility in customising alerts and managing a database system effectively. We can get to all the internals and look into detailed trace information to get more analysis on the alerts that we receive in MySQL and Postgres. Additionally, the collection of all the required statistics from an open source system helps in speedy identification and resolution of some of the major database issues.

**Q** Why would there be a need to reach out to Mafiree for deploying a database solution?

At Mafiree, we come with vast MySQL and Postgres expertise across verticals like e-commerce, logistics, travel and VAS, among others. We also manage the environments of leading e-commerce clients who require high uptime and faster response times. Mainly, we use our open source expertise to help the clients in scaling up their systems and ensuring that they get better throughput and response times, even during peak times.

We help clients implement the latest features that will benefit their current and future data requirements. We provide managed services for MySQL database server solutions such as community versions of MySQL, MariaDB, Percona (XtraDB engine) and TokuDB. Also, we manage large farms of NoSQL databases like MongoDB, CouchDB and Cassandra.

Mafiree comes with eight years of experience in implementing and managing database solutions and has the capability of managing hundreds of database servers across various technologies. We also possess very strong skill sets, with the average work experience of our employees being over eight years in database management.

**Q** Apart from database management, you also provide open source operating system support. What are the parameters you finetune on Linux platforms like Red Hat, CentOS or SUSE?

Our team is capable of performing any kind of performance improvement activities on Linux platforms. To be very specific, we do kernel tuning, TCP stats tuning and disk optimisation. Apart from these processes, we can bring in best practices in patching, upgrade and audit cycles. We also perform orchestration, as well as the recent advancements in arranging, managing and coordinating of systems, middleware and services.

**Q** What is the USP of Mafiree in the world of IT solutions providers?

Managing an in-house database team comes with a lot of practical difficulties. You need to ensure there are sufficient team members when there is an issue or during peak traffic times and also address attrition issues. However, working

**About Mafiree**

Mafiree provides managed services for the MySQL database server solutions such as community version of MySQL, MariaDB, Percona and Postgres, as well as NoSQL offerings like MongoDB, Cassandra and CouchDB. It also delivers services to manage and monitor Web and application servers, mail servers, caching servers and offers developmental support for the LAMP stack. Basically, it supports everything on top of the LAMP stack. It helps customers to design and re-engineer open source solutions to suit their various changing requirements. It also helps customers to benchmark the solution that is proposed. Apart from all this, it enables customers to migrate their database from Oracle to MySQL/Postgres, from MsSQL to MySQL/Postgres, from DB2 to MySQL/Postgres, from RDBMS to MongoDB/CouchDB/Cassandra or from MongoDB/CouchDB/Cassandra to MySQL/Postgres.

with Mafiree helps in eradicating these difficulties by ensuring that the required expert resources are available for different types of issues. We ensure that there is business continuity for the customers, and they can put their effort into improving their services or products, leaving the infrastructure and database management to us.

**“** Open source offers a lot of flexibility in customising alerts and managing a database system effectively. **”**

**Q** How does Mafiree effectively scale up the overall business architecture?

Mafiree plays an important role in getting the right kind of architecture in place for its clients. We can provide a fault-proof, scalable, consistent and high-performing IT solution for our clients. We also possess a young and enthusiastic team, with a keenness to learn and implement new technologies to improve the business operational efficiencies of our clients. We have done lots of reengineering activities that have resulted in performance gains for our customers.

**Q** Which part of the globe do you serve the most?

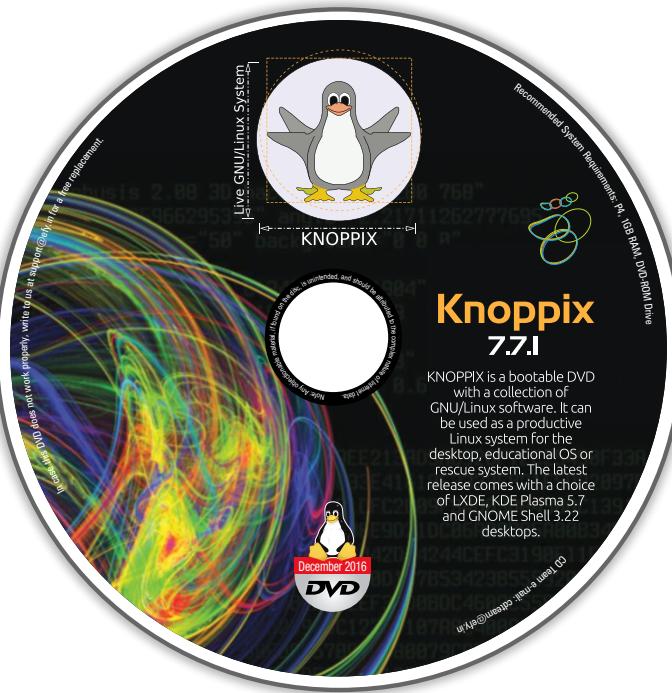
Most of our customers are from Delhi, Gurgaon, Noida, Bengaluru, Chennai, Hyderabad, Kolkata and Mumbai. We also have some clients from Australia, Dubai, UK and the US.

**Q** Last of all, do you plan to stick to open source solutions for future growth as well?

Yes, of course. Open source solutions are gearing up, and there are huge deployments being made on open source platforms. Big companies like Oracle and Microsoft are also venturing into this space, which shows the inevitable need for open source. We are increasing our strengths in the open source stack to ensure that we can cater to all the needs of our customers. 

# DVD OF THE MONTH

Experience Linux in live mode.



## What is a live DVD?

A live CD/DVD or live disk contains a bootable operating system, the core program of any computer, which is designed to run all your programs and manage all your hardware and software.

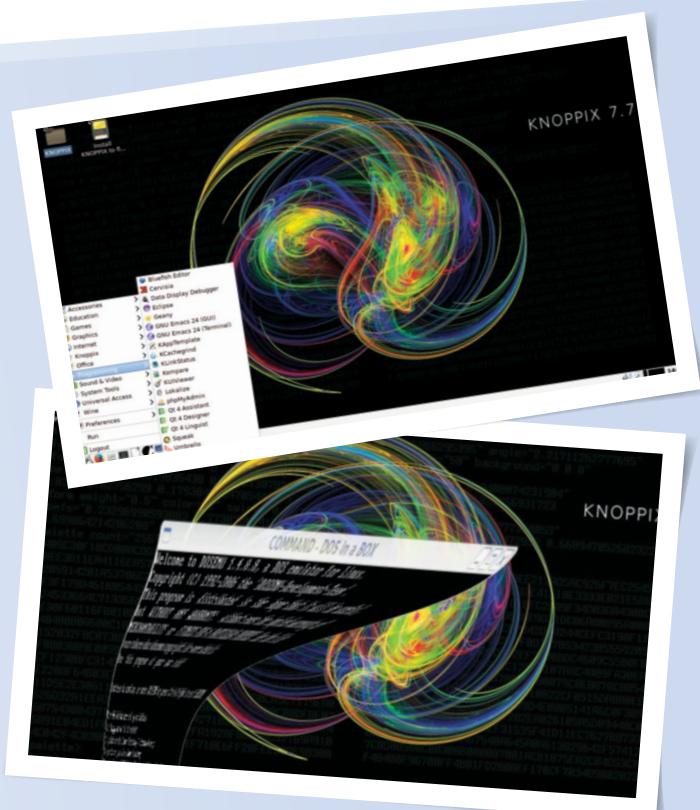
Live CDs/DVDs have the ability to run a complete, modern OS on a computer, even without secondary storage, such as a hard disk drive. The CD/DVD directly runs the OS and other applications from the DVD drive itself. Thus, a live disk allows you to try the OS before you install it, without erasing or installing anything on your current system. Such disks are used to demonstrate features or try out a release. They are also used for testing hardware functionality, before actual installation. To run a live DVD, you need to boot your computer using the disk in the ROM drive. To know how to set a boot device in BIOS, please refer to the hardware documentation for your computer/laptop.

## KNOPPIX 7.7.1

KNOPPIX is an operating system based on Debian and is designed to run directly from a live DVD. It is one of the oldest Linux distros that runs live. The first edition of the live CD was released way back in September 2000.

This is a bootable DVD with a collection of GNU/Linux software. It can be used as a productive Linux system for the desktop, an educational OS, rescue system, or adapted and used as a platform for commercial software product demos. The latest release comes with a choice of LXDE, KDE Plasma 5.7 and GNOME Shell 3.22 desktops.

The minimum hardware requirements for running this DVD are an Intel/AMD-compatible processor (i486 or later) with 512MB RAM and a bootable CD-ROM drive.



# advertising mantras

A good advertisement is one which sells the product without drawing attention to itself.

- David Ogilvy

If your advertising goes UNNOTICED everything else is ACADEMIC.

- William Bembac

Good advertising does not circulate information. It penetrates the public mind with desires and belief.

Considers Itself Immune to The Necessity for Advertising Sooner or Later Finds Itself Immune to Business.

Creative without strategy is called ART. Creative with strategy is called ADVERTISING.

The best advertising should make you nervous about what you are not buying.

- Mary Wells Lawrence

## Doing Business Without Advertising is Like Dancing in The Dark. You Know What You're Doing, But Nobody Else Does

-Stuart H. Britt

1

strategy is called ART. Creative with strategy is called ADVERTISING.

The best advertising should make you nervous about what you are not buying.

- Mary Wells Lawrence

Many a small thing has been made large by the right kind of advertising.

Branding is what people say about you when you are not in the room

If it doesn't sell, it isn't creative.

-Mark Twain

ELECTRONICS FOR YOU

OPEN SOURCE FOR YOU

ELECTRONICS BAZAAR

For more advertising mantras, visit: <http://efly.in/advertising-mantras>

# INDIA ELECTRONICS WEEK

March 2-4, 2017. BIEC. Bengaluru

[www.efyexpo.com](http://www.efyexpo.com)

An  
**EFY GROUP**  
EVENT

Exhibition &  
Knowledge Partner  
**ELCINA**

India's  
Electronics  
Manufacturing  
Show

## Electronics For You **expo**

MAKE. **BUY. SELL. INVEST.**

March 2-4, 2017.  
BIEC. Bengaluru

For more information, talk to us at +91-11-40596605  
or email at [efyexpo@efy.in](mailto:efyexpo@efy.in)