STM JOURNALS
Journal of Artificial Intelligence Research & Advances
ISSN: 2395-6720 (Online)
Volume 5, Issue 3
www.stmjournals.com

A Study of Various Machine Learning and Deep Learning Libraries/Platforms with Their Usage Perspective

Tushar Champaneria[1,]*, Ketakee Nimavat[2], Maitrik Shah[1], Sunil Jardosh[3]
[1]Assistant Professor, Department of Computer Engineering, L.D. College of Engineering, Ahmedabad, Gujarat, India
[2]UG Student, Department of Computer Engineering, L.D. College of Engineering, Ahmedabad, Gujarat, India
[3]Principal Software Engineer, Progress Software, ILAB's Centre, 4th Floor, #18, Madhapur, Hyderabad, Telangana, India

Abstract
Learning is a complex multi-faceted phenomenon. While we do not understand how learning happens entirely, we understand it well enough to devise models that excel at narrow domain learning. Machine learning and deep learning enable us to create models that learn and implement that learning. There is an abundance of tools that achieve the same goals in different levels of abstraction, amount of work done and flexibility to tailor learning methods. However, they are not one-trick ponies and most of the times do not prove to be suitable for all problem scenarios involving machine learning and deep learning. Often the distinction between platforms and libraries remain as: platforms provide an abstraction of the techniques and allow easy implementation of machine learning at scale whereas libraries provide methods to build tailored solutions and often will involve coding from scratch. Depending on who is building, the amount of time and resources available, a library as well as a platform can help one get to the solution. In this study the aim, hence, is to analyze various available tools for building the models in order to make it easier to decide which tool to use for a given scenario.

Keywords: *Machine learning, deep learning, libraries, platforms*

****Author for Correspondence*** E-mail: tac@ldce.ac.in

INTRODUCTION

Machines have always been the cursors of ease, making everyday tasks easier and revealing new horizons to be pursued with renewed vigor as they add a new capability to our diverse skill set. Until the early 2000s, successful advances have come often only for procedural tasks, tasks which can be written down as a rule set or a sequence of instructions, but with the advent of machine learning, the ability to accomplish dynamic and not so straightforward tasks such as recommendations and predictions have become not only easy but also ubiquitous. Machine learning is when the computer can learn about the system without being explicitly programmed for each case. It refers to the use of statistical and other methods to pursue learning. The learning mentioned here is an estimation of the behavior of a given system. This learning was first proposed to be done mathematically, where the meta data of the system helps predict system behavior [1–3]. Later, various methods inspired by nature such as neural networks were proposed. Today, almost every new service coming up has some sort of machine learning formula it is using. Right from adjusting AC temperature to driving cars, machine learning has been seen to fit in. Not only that, things such as a medical lookup website such as WebMd and virtual assistants that were previously thought devices of 2050 are now not only in existence but are also common place. Learning is an important milestone in achieving functional Artificial Intelligence, although, whether intelligence similar to artificial general intelligence is accessible or not remains a thing to be seen in the decades that come. But meanwhile, what cannot be ignored is the power machine intelligence brings to everyday lives. Creating machine learning models has become simpler over time as well. A field that was previously open to a clique of scientists with advanced mathematical and computational science knowledge is now open to anyone with programming experience for experimenting with its applications; the research has become more accessible as well but still requires a depth of mathematical knowledge. Building machine learning applications has become more accessible partly due to the increase in software and libraries providing an abstraction from the nitty-gritty of machine learning and giving a black box to enter the data in instead. Coding machine learning models from scratch has become simple as well owing to the packages and libraries available in the general-purpose languages and the emergence of new languages that make coding the complex math involved in machine learning simpler. It has now become common place to execute a machine learning algorithm on the data one has. With the plethora of tools available for implementing machine learning, it is still often confusing to decide the suitability of the tool for a task for a variety of reasons ranging from insufficient knowledge of the capability of the tool to insufficient knowledge of other tools. One might be using TensorFlow without awareness of the optimizations it is capable of or one, being a learner who has just read about neural networks, might end up looking for python libraries such as CNTK when all he needs perhaps is a platform like Azure ML studio to understand the big picture.

In this study, we explore various libraries and platforms for implementation of machine learning algorithms. To do so, we first go through the history of machine learning and deep learning in section II the way they grew and their efficiency in solving today's problems. Next, we look at problems where machine learning and deep learning are used and discuss why it is so in section III. In section IV the reason behind the emergence and suitability of various languages for implementing the machine learning and deep learning is discussed. Followed by section V where various libraries of various languages are discussed along with their special features. Along with libraries, platforms and the level of abstraction they provide, cost etc. are discussed as well. Finally, we discuss their usage and layout of the tasks each library is known for solving. We, hence, hope to make it simpler for a user to decide which library to use for each of the possible machine learning problems in light of the information presented in this study.

THE OVERVIEW OF MACHINE LEARNING AND DEEP LEARNING

Machine Learning can be defined as a set of mathematical constructs and algorithms made for learning. This learning happens in terms of probabilities in probabilistic models, in terms of a vector in neural networks and in terms of construction of probabilities of the next branch in decision trees. Machine Learning algorithms access data and draw inferences from it. Like we humans do. Yet, machine learning is not so simple owing to the multiple facets of tasks it involves [4]: task-oriented learning, cognitive simulation, and theoretical analysis.

Machine learning consists of a variety of tasks with a specified set of algorithms suited for each task. Right from regression tasks to language generation and face recognition. Simple mathematical models have proven to be insufficient for various complex tasks such as image recognition. This being because most of the machine learning algorithms that could be implemented given the computational power of yesteryear worked on the conclusions derived on the overall data, they did not work with individual data points or did not remember the details. The adaptation of neural networks came after the computational power was enough to run them on large amounts of data. The neural networks proved to be more powerful in some fields such as image processing for they were capable of doing both, understanding the lower level details and the abstraction.

Neural networks came into the picture in 1957 with the discovery perceptron’s, but the forms of neural networks popular today did not come into being until the 1980s when back propagation, recurrent neural networks and reinforcement learning were discovered. Fast forward to 2017, neural networks has been experimented with and developed to remember information and generate data and are dominant in every field be it translation or image recognition. Deep learning consists of a set of tools that is a subset of machine learning tools. Deep learning is a set of algorithms that help achieve the tasks that machine learning set to achieve. Deep learning is hence a subset of machine learning as shown in Figure 1.

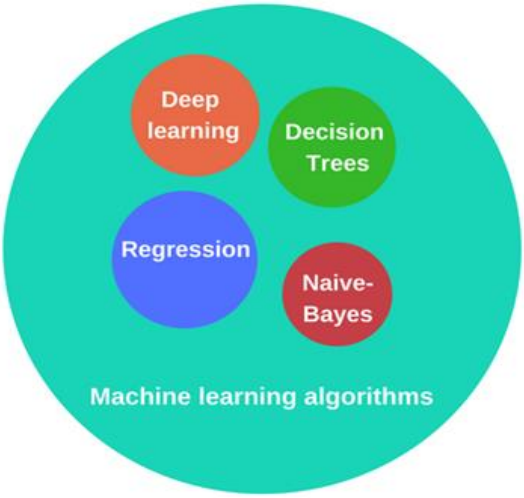

Fig. 1: *Machine Learning and Tools to Achieve Machine Learning.*

Today, neural networks are synonymous to Deep Learning, but deep learning is defined by a set of algorithms and techniques that often involve multiple successive layers that perform feature extraction and transformation. These multiple layers enable the model to learn representations of the input data at various levels usually split from a higher level to lower level features with higher level features being abstract ones and the more complex details being captured in the lower features [5]. Deep learning methods also use gradient descent in some measure. A full-fledged Deep learning model hence is similar to a scaled up neural network which has multiple layers and nodes. The increase in accuracy that deep learning brings comes at the cost of computing power. Due to which, libraries have been developed specifically for deep learning tasks.

Deep Learning and machine learning seem to be complete solutions in themselves; the existence of one seems to invalidate the need for the existence of the second approach. For example, Image processing is a field that involves multiple approaches to it, the statistical machine learning approach and the deep learning approach. The statistical machine learning approach often involves derivation of a result based on metadata. This is used in tasks such as face detection, background subtraction etc. Let us take the example of background subtraction in image processing. Background subtraction is needed in use cases such as self-driving cars where the position of the subject changes continuously. Background subtraction can be achieved by both methods: Statistical machine learning and neural networks. In statistical machine learning, Gaussian filtering is one of the methods used and has shown considerable results. Neural Networks can be used for this task as well; they do not rely on statistical metadata and instead understand the picture in layers of abstraction with each layer considering a more detailed view than the previous one. In this case, both approaches provide complete solutions.

However, not all problems have both machine learning and deep learning solutions; some have only deep learning solutions. Take the example of image Description, it would be difficult for statistical methods to take in the inputs as image and text, even if they do, achieving accuracy would be difficult as well. Neural networks with multiple layers and feedback loops, on the other hand, have been seen to accomplish this [6, 7]. When given an image, it is sent through a CNN that consists of a rich encoding of the image. The output of the CNN is sent to an RNN which is trained for generating language. This can be further optimized by processing the picture in parts. Another example where only machine learning is not sufficient enough is a generation of images from a given text; this has been accomplished using GANs [8]. GAN, short for Generative Adversarial Network is a system wherein two networks work together to achieve a system wherein one network is a generator and the other is a discriminator. The job of the generator is to generate samples such that the discriminator cannot tell the difference between real and fake samples, fake being the ones generated by the generator. By doing so the generator learns the system properties. Such robust learning methods are not always possible in statistical machine learning due to which deep learning methods are better suited for real-world problems of language, images, understanding, etc.

SAMPLE SCENARIOS

When one has a bulk of data, intuitively, one can do three things. Analyze the data, predict future outcomes based on the data or fit new data points into classes according to the old data. Let us take the example of an e-commerce site selling books where a user brought three storybooks and three textbooks. Figure 2 shows a diagram of how a user's data is used for different tasks independently. We see the questions asked for each task and the algorithms that can be used. The major three tasks that are often performed on data such as that of an e-commerce site are as follows:

Classification: When the category is determined according to pre-determined categories and is trained on a labeled data set, the task is called classification. On the other hand, if data is grouped as it comes, without any prior information about the classes or without the labels of the classes, it is called clustering. Similar data points are grouped together as they are encountered and if a new point does not lie in any of the given groups, a new cluster is formed. For example, classify a new user into an age group based on his purchases and other people who have purchased the same things. Hence, in classification, existing data points are grouped together *into* classes. Each point in a class is similar for all fields to the other points in that class. Based on the books a user bought, the user can be classified as to which grade he belongs to or how old he is.

Prediction: It involves extrapolating or interpolating based on given data points, i.e., Suggest the next thing the person would probably like to buy. Given the purchases of a person, their next purchase can be predicted based on what other people bought along with or after these items. Based on the books the user bought, it can be seen which other books people bought along with them and suggestions can be made. For example, perhaps he bought two of the five textbooks that people often buy for a fifth-grade science. Based on this, the other three textbooks can be suggested as well.

Analysis: Buying trends can be analyzed by age, geography, time of the day, etc. This helps in understanding the patterns and deciding marketing strategies. It also helps in reviewing previously employed strategies and analyzes who buys what the most.

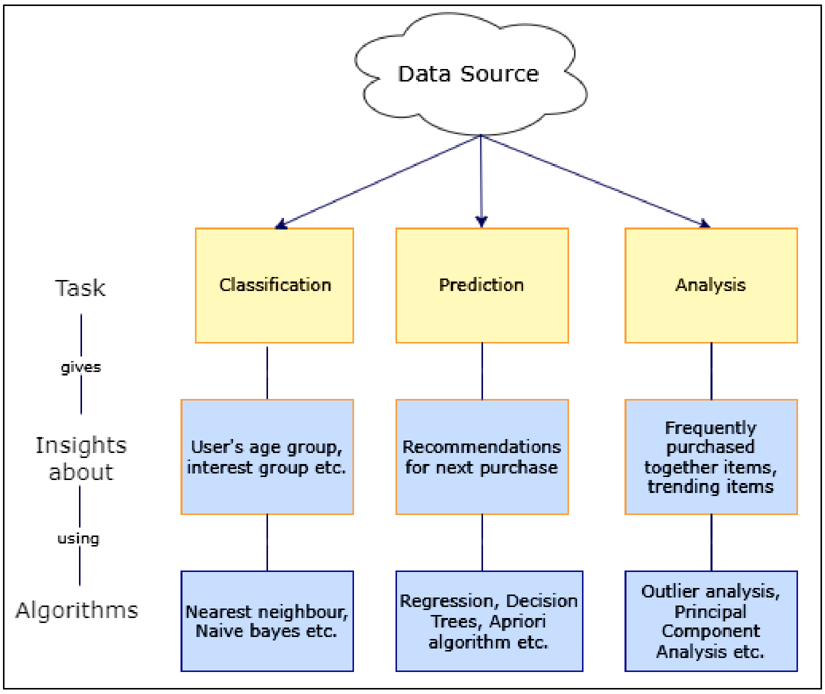

Fig. 2: *Example Usage Scenarios of An E-commerce Site Data.*

As seen in the previous section, various problems such as background subtraction or stock price prediction may be solved by both: machine learning and deep learning, while others such as image description may be specifically solved by deep learning. For the problems that have solutions that give acceptable results using both approaches which approach to choose depends upon the computational power available and the level of accuracy needed. As the emergence of ML/AI, development efforts, implementation aspect differs as per different languages. It also depends upon data available, time required for processing, platform the application is built upon and the languages available for using, whether the language being used is a recent one focused on machine learning or is a general-purpose language like C/C++ or Java. These implementation aspects are discussed in the coming sections.

IMPLEMENTATION ASPECTS OF MACHINE LEARNING AND DEEP LEARNING

The idea of which approach to choose based on the platform begs the question, cannot all platforms execute all machine learning algorithms? Why do so many new languages emerge when the old ones such as C++ are claimed to be extremely powerful and are indeed the base of the new emerging languages? A new language emerges when the requirements and the users of the language change. As the use of languages for application purposes increased, coding in basic languages such as C/C++ became tiresome [9]. High-level languages such as Java became popular because of its already available libraries which made coding simpler. These new languages have inbuilt libraries for things that are used often but have no inbuilt implementation in the existing language. The existing language imports a third-party library for the purpose which increases complexity and dependencies. Third party libraries also do not provide flexibility of implementation. On the other hand, the newly emerged language not only has an inbuilt library, but it also has efficiently implemented version of the same. The new languages are faster and more efficient, for example, Go is closer to the hardware and are more similar to regular English. The emergence of various languages for machine learning took place due to reasons similar to the ones mentioned above. Machine learning involves scale, millions of computations to get one pattern. This kind of computation is hard to formulate in C++ [9]. The algorithms require data structures that optimize CPU usage, something that perhaps not all programmers are capable of doing. Python is perhaps the most popular language for machine learning these days owing to its high level and ease of use. Along with that, it is simple, elegant consistent and has a little bit of everything. Ultimately python is written in C/C++ but doing machine learning directly using lower level languages demands a deep knowledge of those languages in order to create optimized programs. Additionally, C++ has STL and other third-party libraries, but these are not sufficient for the machine learning problems. Instead, while implementing them in python, the data structures that are lacking are already implemented optimally. Along with that, it frees the programmer from the concerns of knowing the complex low-level languages and its paradigms. Especially the newer languages made especially for machine learning; they provide one with an abstraction that achieves exactly what the programmer is looking for in a fuss-free manner [9]. A recent survey by economic hub found that python is the most prioritized language in development for machine learning, data science followed by R, C/C++, Java and JavaScript in that order. Languages, such as Julia, SCALA, MATLAB held minorities [10].

Though above-stated languages are used, independent solutions can be developed from scratch but there are libraries are available using which we can kick-start development of any project. Libraries contain prebuilt functions which are something special purpose. For example, OpenCV is a special purpose library for computer vision; NLTK and Textblob are explicitly for text processing. On the other hand, Scikit-Learn and Theano are general purpose machine learning libraries. The decision to use a language and the library in that language depends upon the requirement of the developer and resources available at hand. Also, some languages and tools such as

MATLAB and Weka are used primarily for prototyping and quick implementation. In the next section, this is explored in deep in the form of major libraries and their use cases.

LIBRARIES AND THEIR USAGE

The library contains a set of functions that are frequently used for different purposes. There exist a number of libraries for ML programming having different usages. In this section, we have explored the most popular ML libraries and their usage.

Libraries

Deeplearning4j [11]: It is an open source deep learning library for Java which works on GPUs and CPUs. It includes implementations of the restricted Boltzmann machine, deep belief net, deep autoencoder, stacked denoising autoencoder and recursive neural tensor network, word2vec, doc2vec, and GloVe. It is used in the finance industry for fraud detection, automotive industry for failure detection, for risk detection in areas of security and finance, recommendation engines for social networks and e-commerce, sentiment analysis for reputation management, social media and CRM.

Gensim [12]: Gensim is an open-source vector space modeling and topic modeling toolkit in built using SciPy, NumPy and Cython in Python. It contains models for tf-idf, random projections, word2vec and document2vec algorithms, Hierarchical Dirichlet Processes (HDP), Latent Semantic Analysis (LSA) and Latent Dirichlet allocation (LDA), including distributed parallel versions. It has been used by Rare Technologies, for NLP applications, by Mindseye for finding, by Cisco for large-scale fraud detection similarity between legal documents, by Amazon for finding similarities between documents and many other corporations for text related applications.

Keras [13]: Keras is a cross-platform python library written specifically for neural networks. It can be integrated with or used on top of MXNet, Deeplearning4j, TensorFlow, Theano. The paradigm of this library revolves around modularity, mineralization and extensibility. Keras is also now supported in TensorFlow's core library. While other libraries are themselves implementations, Keras on the other hand is a framework that provides abstractions that enable implementation of other back-end libraries mentioned above.

MXNet [14]: MXNet is a framework that is supported by multiple programming languages such as C++, R, Python, JavaScript, Julia, Matlab, Go, etc. It is the library that supports the most languages, supports GPUs as well as CPUs and has the support of major cloud providers as well, especially Amazon where MXnet is the framework in AWS. The major features are flexibility, multiple languages, portability, scalability. Because of these features, this library is supported by various organizations such as Baidu, Microsoft, Intel, MIT, Carnegie melon, etc. MXnet is used for face detection, Optical character recognition, machine translation and other such deep learning applications.

Microsoft CNTK [15]: It is an open source toolkit for deep learning created by Microsoft. It is the only toolkit that can scale beyond two machines. While it is a toolkit, it can also be used via Python, C++ libraries available. It is used as a backend with Keras in Python. It has been used for machine translation, image processing and other such deep learning tasks.

OpenNN: OpenNN is a cross-platform open-source library written in C++. It has a deep architecture for supervised learning in the form of layers of non-linear processing units which enables the creation of neural networks that have universal approximation properties. It has data mining functions but lacks a GUI which can be made up for by third-party visualization tools. OpenNN also has a companion tool called Neural Designer which provides an interface and abstraction for performing advanced analytics tasks.

Torch [16]: Apart from being a machine learning library, torch doubles as a scientific computing framework and a scripting language. It is based on Lua and majorly has algorithms for deep learning. It has major packages as torch and nn with other various packages that add the functionalities of parallelism, image processing etc. Torch is used by Facebook, IBM, Yandex. Like TensorFlow, torch also has android and IOS compatibility. It is mostly used for models that require heavy dataflow such as neural networks.

TensorFlow [17]: Created by the Google brain team, it is an open-source machine learning library designed majorly for neural networks but is capable of a variety of machine learning tasks. It is the fastest growing machine learning library. It has a version called TensorFlowLite which is designed for android development. TensorFlow has been used for speech recognition, text summarization, translation, image processing by various corporations including Google.

Caffe [18]: This is a deep learning framework written in C++ and is implemented in python. It was developed in UC Berkley and is focused on neural networks and image processing. GPU based acceleration is supported via cuDNN (NVIDIA). This library is often used in academic research projects and startup prototypes. It is used with an integration of Keras.

Theano [19]: This is an open source cross-platform library developed by the University of Montreal. It is a python-based library which focuses on efficient representation and compilation of computations on CPUs and GPUs. Theano provides Tensors for computation which enables it's optimized under the hood calculations. Because of this feature, it often gets compared to TensorFlow. It is used especially in use cases involving neural networks.

Dlib [20]: This library is being developed since 2002. It is a cross-platform library being written in C++. Unlike other libraries, it derives its paradigm from a design by contract and component-based software engineering. It hence resembles a collection of software components. In machine learning, it contains software components like any general-purpose programming language: networking, data structures, threads, GUI. But on top of that, it also has the components that a machine learning library would have: machine learning, image processing, numerical optimization, data mining, Bayesian networks, etc.

H2O [21]: While most of the libraries have been focusing on machine learning, this is a software made for big data analysis and runs on Apache Hadoop. It is used for exploring and analyzing datasets in cloud computing and is compatible with the majority of browsers.

Scikit-learn [22]: Scikit-learn [23] is made for python and supports various machine learning tasks such as classification, clustering, regression, k-means, random forest, etc. It interoperates with Numpy and SciPy and is written in Cython. It is used by companies such as Evernote, booking.com, Spotify, change.org, and many more [24].

MLPack [25]: It is a C++ library that focuses on speed, scalability and ease of use. It provides an abstraction that makes it easy to use for beginners and at the same time takes advantage of features of C++. It is open source and is built on top of Armadillo library. It is supported by Georgia Institute of Technology and includes machine learning algorithms such as hidden Markov models, Logistic regression, Naïve Bayes Classifier, principal component analysis, nearest neighbor algorithms and so on.

Natural Language Toolkit (NLTK) [26]: It is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for various languages. Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania has developed. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

PaddlePaddle (PArallel Distributed Deep LEarning) [27]: It is an open-source deep learning platform. It is easy to use, efficient, flexible and scalable deep learning platform, originally developed by Baidu scientists and engineers for the purpose of applying deep learning to many products at Baidu. Concentrated towards an efficient deep learning, PaddlePaddle has a deep learning framework named ‘Fluid’. Fluid is said to be similar to PyTorch and TenserFlow Eager

Execution, which describes "process" of training or inference using the concept of a model. Efforts have been taking place to push Fluid towards the directions of a compiler and a new programming language for deep learning. PaddlePaddle has a binary floating-point data type that occupies 16 bits in memory 'float16'. It is half the size of traditional 32-bit single precision format (float).

Chainer [28]: It is an open source deep learning framework written purely in Python on top of Numpy and CuPy Python libraries. The development is led by Japanese venture company Preferred Networks in partnership with IBM, Intel, Microsoft and Nvidia. Chainer supports various network architectures including feed-forward nets, ConvNets, recurrent nets and recursive nets. It also supports per-batch architectures. Forward computation can include any control flow statements of Python without lacking the ability of backpropagation. It makes code intuitive and easy to debug.

Table 1: *Details of Machine Learning Libraries*

Sr	Library Name	Operating System	License	GPU Support	Written In	Language Interface	Use Cases
1	Deepleanring4j	Linux, Mac, Windows, Android	Apache 2.0	Yes	Java	Java, Python, Scala, Closure	Deep Learning
2	Genism	Linux, Windows, MacOS	LGPL	Yes	Python	Python	Information Retrieval, Natural Language Processing
3	Keras	Windows, OSX, Linux, Debian, Fedora, Ubuntu, openSUSE	MIT	Yes	Python	Python, R	Neural Networks
4	Mxnet	Windows, Linux	Apache 2.0	Yes	C++, Python, R, Julia, JavaScript, Scala, Go	Python	Deep Learning
5	CNTK	Windows, Linux	MIT	Yes	C++	Python, C++, Command Line	Machine Learning Deep Learning
6	OpenNN		LGPL	No	C++	C++	Neural Networks
7	Torch	Linux, Android, Mac OS, IOS	BSD	Yes	Lua, C, CUDA, C++	Lua	Machine Learning/Deep Learning
8	TensorFlow	Linux, Mac OS, Windows, Android	Apache 2.0	Yes	C, C++, Ccuda	Python (Keras), C/C++, Java, Go, R	Machine Learning
9	Caffe	Linux, MacOS, Windows	BSD	Yes	C++	Python, MATLAB	Deep Learning
10	Theano	Ubuntu, Mac OS, CentOS6, Windows, NVIDIA Jetson TX1, AWS, Docker, Gentoo	BSD	Yes	Python	Python	Deep Learning
11	Dlib	Windows, MacOS, Ubuntu	Boost Software License	Yes	C++	C++	Deep Learning
12	H2O	Linux, MacOS, Windows	Apache 2.0	Yes	Java, Python, R	Python, R	Big Data Analytics
13	Scikit-Learn	Linux, MacOS, Windows	BSD	No	Python, Cython, C, C++	Python	Machine Learning
14	Mlpack	Windows, Ubuntu	BSD	No	C++	C++	Machine Learning
15	Paddle Paddle	CentOS 6 or later, Ubuntu 14.04 or later, MacOS 10.12 or later	Apache 2.0	Yes	C++	C++, Python: 2.7.x	Deep Learning
16	Chainer	Ubuntu 14.04+, CentOS 7+	MIT	No	Python	Python 2.7.6+, 3.4.3+, 3.5.1+, 3.6.0+	Complete implementation of deep learning
17	NLTK	Windows, MAC OS, Linux	Apache 2.0	No	Python	Python versions 2.7, 3.4, 3.5, or 3.6	Natural Language Processing

So far general-purpose machine learning libraries have been discussed however, certain libraries are known for specific tasks. For example, Genism is known for natural language processing. Table 1 shows the details of machine learning libraries with their use cases. Hence in the coming paragraphs, the user of machine learning development tools, levels of abstraction and well-known usage of various libraries are discussed.

Usage

As shown in Figure 2, various machine learning abstraction levels allow different people to use machine learning. Executives and knowledge workers will only want the results of the machine learning service, in-house engineers who are familiar but not fluent with machine learning will want to be able to play with the models but may not necessarily want to build them from scratch. Core machine learning engineers and researchers, on the other hand, will want to go beyond the black box system and tinker under the hood. Apart from abstraction, the performance of libraries and platforms matters greatly as well. As discussed in the previous section, there are a variety of libraries and platforms available. Each library has its own feature, it may be competent at a certain task such as neural networks and others may not be. These libraries can be benchmarked for a given task in order to decide which one to choose. An example of benchmarking on GitHub [29] shows benchmarking of various open source libraries of R, Python, XBooost, H2O, etc. for common machine learning tasks such as regression, classification etc. The methodology may be used to obtain benchmarking for one's one environment, depending upon which appropriate models may be chosen.

For further discussion on usage, we look at some real-world problems and the algorithms used in them. Following which, Table 2 shows various libraries and the problems for which they are used along with the list of problems for which it can be used. Built-in function support refers to the availability of a prebuilt interface for machine learning models that would make implementation easy for programmers. A low on which signifies that is often used for building the models from scratch. As machine learning can be applied to broad categories of domains, here we have discussed few popular problem domains.

Image processing: Problems such as Image recognition and image classification fall under this domain. The algorithms used often are pattern recognition, clustering, Classification etc. which are implemented using neural networks, K-means clustering, Naïve-Bayes classification, etc. [30, 31].

Security: Anomaly detection, Fraud detection, intrusion detection. While not a specific machine learning field, security has various machine learning applications which are solved using image processing, computer vision, Regression etc. The specific problem pertaining to these areas are automatic doors, intrusion detection, and anomaly detection [32–34].

System analysis: This field pertains to the analysis of the behavior of the system to detect potential threats. Problems such as flaw detection in engines, wear and tear prediction of engines and vehicle parts, outlier analysis fall in this field. Most of them use algorithms such as regression, K-means, K-NN, Support Vector machines, Principal component analysis, etc. [35–37].

Natural language processing: Natural language processing and natural language understanding are emerging fields. Various problem areas that are being worked upon include: conversational agents, subject detection for processing queries, question answering systems, Translation, Summarization, smart Replies etc. The algorithms used for these include neural networks, probabilistic methods such a Naïve-Bayes, support vector machines, Latent Dirichlet Allocation, etc. [38–43].

Gaming: Gaming is often used to test methods of learning such as reinforcement learning and transfer learning. Typical areas are game landscapes generation and teaching an agent to play games. These are achieved using GANs and reinforcement learning [44].

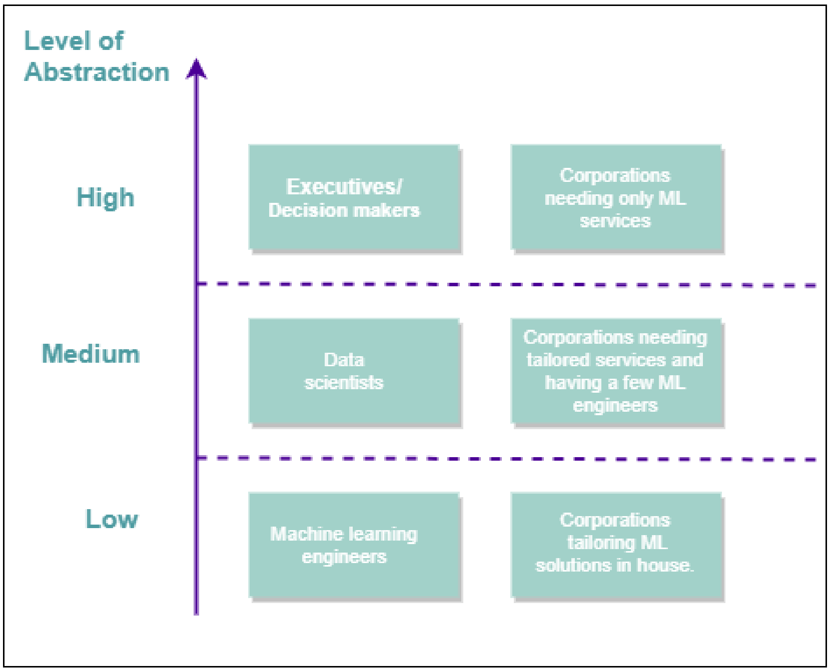

Fig. 3: *Levels of Abstraction vs Usage Scenario of Machine Learning Tools.*

Table 2: *Libraries and Their Usage.*

No.	Library	Suggestive Usage	Built-In Function Support
1	Deepleanring4j	Fraud Detection, Security, Anomaly Detection, Recommendation Engines	High
2	Genism	Text Mining, Similarity Detection, Fraud Detection, Recommendation Engines, Document Classification	High
3	Keras	Image Classification, Auto encoders,	High
4	Mxnet	Image Processing, Natural Language Processing, Generative Algorithms, Auto-Encoders	High
5	CNTK	Image Processing, Natural Language Processing, Language Understanding, Video Processing, Image Processing, Classification	High
6	Opennn	Modeling, Pattern Recognition, Time Series Prediction, Regression	High
7	Torch	Image Processing, Natural Language Processing	High
8	TensorFlow	Speech Processing, Image Processing, Natural Language Processing, Time Series Analysis, Video Processing, As A Backend for Keras	Low
9	Caffe	Classification, Image Processing, Speech Processing	High
10	Theano	As A Backend for Keras for computational efficiency	Low
11	Dlib	Classification, Regression, Prediction, Clustering, Reinforcement Learning, Image Processing	High
12	H2O	Time Series, Pattern Detection, Churn Modeling, Fraud Detection, Word2vec, Prediction Tasks	High
13	Scikit-Learn	Computer Vision, Security Applications, Recommendation Systems, Clustering, Feature Selection, Principal Component Analysis	High
14	Mlpack	Classification, Regression, Clustering	Low
15	PaddlePaddle	Deep Learning	High
16	Chainer	Complete implementation of deep learning	High
17	NLTK	Natural Language Processing	High

Recommender systems: Recommendations systems are widespread these days. They are present in e-commerce, search results and are used for targeted advertising [45–46].

Finance: Risk analysis and anomaly detection/fraud detection, are perhaps the most common applications of finance. Regression, K-means and K-NN clustering methods used for it. Behavior analysis is an important factor in finance as well [47–48].

To summarize, while all of the above-discussed libraries are general purpose libraries, particular libraries have grown to be known for specific tasks because of their accuracy or ease in performing that task. It is hoped that the above matrix in Tables 1 and 2 help the reader chose which library to use given their task and scenario. For reduced hassle and a readymade solution, machine learning platforms provide enough services to quickly prototype solutions for business requirements. Hence, in the coming section, platforms are discussed along with their characteristics such as abstraction, algorithms provided, etc.

PLATFORMS

Platforms provide a means to achieve major machine learning and deep learning tasks. Some platforms such as the Azure ML studio provide a drag and drop interface with prebuilt algorithms. Whereas others such as the Google cloud platform lets one tinker with the code of the models being used. Apart from being easy to use, platforms help achieve scalability since they are on the cloud for the given machine learning application.

Table 3 shows an overview of the following platforms along with their OS, Data source (where the data has to be stored while using the platform's services), Algorithms supported by the platform, whether the platform suggests algorithms based on data (Algorithm suggestion), if it has prebuilt machine learning models that are ready for application (Prebuilt models) and their configurability (how much can the algorithm be tweaked to suit one's needs).

Amazon Machine Learning [49]: Amazon Machine Learning promises to be a platform for coders of all skill levels. It provides pre-built models for various algorithms, data transformation tools, model evaluation and interpretation tools and APIs for batch and real-time predictions. It is a pay per use plan and provides an entire black box for algorithms although pricing varies as per regions. It has a binary multiclass classification, regression. Amazon has a variety of services under its Amazon AI wing. It has other branches namely Polly and Lex along with core machine learning services in amazon machine learning. Amazon has other modules such as Rekognition, Polly, Lex allow the user to incorporate image processing, speech generation and an Alexa like conversational interface in voice and text. It combines its platform service and its machine learning service to provide complete machine learning Infrastructure as a service.

IBM BlueMix [50]: IBM BlueMix is primarily an analytics platform with machine learning services provided by Backend IBM Watson services. Watson provides a wide array of services including conversational agents, translation services, search systems, image processing application, tone analyzing, natural language understanding etc. IBM Watson has the most diverse set of services among services provided by platforms.

Azure ML Studio [51–53]: Azure ML Studio provides a drag and drop interface with prebuilt models along with built-in packages for R and Python to support custom coding. On top that, Microsoft services such as computer vision, emotion API, LUIS provide a wide array of services in the form of an API that users can use. Apart from that Microsoft also provides Machine Learning Workbench, Azure Machine Learning Experimentation Service and Machine Learning Model Management Service. Machine Learning Experimentation Service is made for easy deployment of experiments and to enable scaling from local machines to the scale of hundreds of GPUs. The workbench provides.

Google Cloud Platform [54]: The Google cloud platform aims to provide machine learning at scale. The underlying framework is that of TensorFlow and is heavily based on neural nets. Apart from that, one can upload their own models or use one Google's prebuilt models. It has inbuilt APIs for image analysis, video analysis, speech analysis and text analysis and translation. Google Data Studio is another platform from Google which provides data visualization tools for creating reports.

Table 3: *Details of Machine Learning Platforms.*

Sr.	Platform	OS/Cloud	Data	Algorithm Support	Algorithm Suggestion	Prebuilt Models	Configurability
1	Amazon Machine Learning	Cloud: Amazon EC2 Cloud	AWS S3, EBS, EFS	Classification and Regression. Rekognition, Polly, Lex.	Yes	Yes	Low
2	Google Cloud Platform	Google Cloud Platform	Google Cloud Storage	APIs Available for Text to Speech, Image Analyzer, Video Analyzer, Speech and Translation	Yes	Yes	High
3	Weka	Windows, OSX, Linux	Own Data, small quantities	For Data Analysis: Classification, Clustering, Regression	No	No	Low
4	Microsoft Azure Machine Learning	Cloud: Microsoft Azure	Small data sets can be imported but otherwise on Microsoft servers	Algorithms: Classification, Regression, Analysis, Text Analysis, Recommendation Systems, Deep Neural Networks, APIs for: Text Analysis, Video Analysis, Image Analysis, Bing Search, Emotions from Images etc.	No	Yes	Low
5	IBM Bluemix	Cloud: IBM Bluemix Platform	IBM Bluemix	IBM Watson Services Like: Watson Conversation, Visual Recognition, Virtual Agent, Knowledge Discovery, Speech, Language, Empathy	No	No	High

Alibaba Cloud [55]: One of the services provided by Alibaba cloud is its 'Machine Learning Platform For AI'. It is built on the full-fledged algorithm application system of Alibaba Group. It is said to offer asynchronous and parallel communication which directly results in supporting the training tasks containing tens of billions of features. It also supports dynamic feature control that enables addition and removal of model features at any time which is, in fact, proving out to be a powerful tool for CTR prediction scenarios. With a highly interactive user-interface and a vast variety of machine learning algorithms available on-the-go, Alibaba Group's platform has had a successful implementation in 'Sina Weibo', a leading social media platform in China with 165 million daily active users and 376 million monthly active users. Its simple architecture is divided into 4 parts: Infrastructure, Computing Framework, Model and algorithm, Business application layer. The machine learning modules included in the platform are Data Preprocessing, Statistical analysis, Machine Learning, Deep Learning, Time series, Text analysis.

Hence, as it can be seen, the amount of abstraction and control offered varies from platform to platform. Which platform to use depends upon the kind of people using it and the task it is being used for. If there are no machine learning engineers or researchers available, a corporation might choose to use the Azure machine learning or IBM Bluemix for implementations of basic algorithms such as classification or regression.

Platforms and libraries are both for their specific scenarios. It is not mandatory to use one or the other. Decision makers might want to extract information about patterns and decide the next move. They would not want to deal with optimizing algorithms or choosing step-sizes. An in-house machine learning team, on the other hand, might want to optimize machine learning algorithms and experiment with the construction of layers in order to build solutions for the company. Platforms such as Google cloud platform that provide the flexibility of writing one's own code can be used and libraries such as TensorFlow will be included then. On the other hand, Amazon Machine Learning can be used by any organization to provide machine learning applications rather than coding the applications from scratch. Giants such as Amazon, Google, Microsoft and IBM all cover almost the same areas of services, however, each one has its own expertise. Amazon goes to an extra length for photos containing humans by giving details such as "eyes closed/open". Microsoft on the other hand provides extensive metadata about the pictures including fields such as background colour, dominant colour, key tags and even a caption that very well captures the essence of thee image. Watson has extensive services with added modules such as Discovery News, Knowledge Studio and Document Conversion that allow the user to embed dynamic news, mine unstructured text for knowledge without coding and automatic conversion of documents into various languages and file formats. Microsoft's LUIS can be easily integrated with the Microsoft Bot framework to provide a Chabot complete with Natural Language Understanding and Analytics.

Hence, each platform has its set of services that it is well known for. While every platform provides the same services in general, they differ in depth of information returned and in the functions the service provides. Therefore, which platform to select is dependent upon the amount of information required in the particular use case. Secondly, it also depends upon the existing infrastructure, if a system is completely built using Google Cloud Platform; it will be easier to integrate Google NLP services than to integrate Microsoft or IBM Watson services. It is hoped that the above discussion proves to be a sufficient guide in helping the reader chose which platform to use.

CONCLUSION

Machine learning and deep learning hence have various available methods for interpreting data. Machine learning except for Deep learning often work on metadata of the data and deal with data as a whole. Deep learning, on the other hand, is capable of capturing the nuances of the instances involved. Both techniques are computationally intensive because of the amount of training data involved, have their own respective libraries that work on their problem types. We have grouped together various libraries and platforms available and the tasks they are good at. We hope this helps the user understand the purpose and prevalence of various libraries/platforms and gives them an insight into what should be used for the task at hand.

REFERENCES

1. Quinlan JR. Induction of Decision Trees. *Mach. Learn.* 1986; 1 (1), 81–106p.
2. Sutton RS. Learning to Predict by the Method of Temporal Differences. *Mach. Learn.* 1988; 3 (1): 9–44p.
3. Aha DW, Kibler DF. Noise-Tolerant Instance-Based Learning Algorithms. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (I).* 1989; 66: 794–799p.
4. Carbonell JG, Michalski RS, Mitchell TM. An Overview of Machine Learning. In: *Machine Learning.* Berlin, Heidelberg: Springer; 1983; pp 3–23.
5. Bengio, Y. Learning Deep Architectures for AI. *Found. Trends® Mach. Learn.* 2009; 2 (1), 1–127p.
6. Vinyals O, Toshev A, Bengio S, Erhan D. Show and Tell: A Neural Image Caption Generator. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2015; 07–12–June, pp 3156–3164.
7. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhutdinov R, Zemel R, Bengio Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. 2015.
8. Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, Metaxas D. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. 2016.
9. Hundt R. Loop Recognition in C++/Java/Go/Scala. Proceedings of Scala Days 2011. 2011; 86 (2): 298–307p. Available from https://days2011.scala-lang.org/sites/days2011/files/ws3-1-Hundt.pdf [accessed September 2018].
10. Voskoglou C. (2017). What Is the Best Programming Language for Machine Learning? [Online] Developer Economics, Towards Data Science. Available from https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7 [accessed September 2018].
11. Team DJD. Deeplearning4j: Open-Source Distributed Deep Learning for the JVM. *Apache Softw. Found. Licens.* 2 2016.
12. Řehůřek R, Sojka P. Gensim–Python Framework for Vector Space Modelling. *NLP Centre, Fac. Informatics, Masaryk Univ. Brno, Czech Repub.* 2011.

13. Chollet F. (2015). Keras. [Online] GitHub. Available from https://github.com/fchollet /keras [accessed September 2018].
14. Chen T, Li M, Cmu UW, Li Y, Lin M, Wang N, Wang M, Xu B, Zhang C, Zhang Z, et al. MXNet : A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems arXiv : 1512 . 01274v1 [Cs. DC] 3 December 2015. 1–6.
15. Yu D, Huang X. Microsoft Computational Network Toolkit (CNTK). *A Tutor. Given NIPS 2015* 2015.
16. Collobert R, Kavukcuoglu K, Farabet C. Torch7: A Matlab-like Environment for Machine Learning. *BigLearn, NIPS Work.* 2011; 1–6p.
17. Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2015.
18. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional Architecture for Fast Feature Embedding. 2014.
19. Bergstra J, Breuleux O, Bastien F. F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y. Theano: A CPU and GPU Math Compiler in Python. *Proc. Python Sci. Comput. Conf.* 2010; Scipy, 1–7p.
20. King DE. Dlib-Ml: A Machine Learning Toolkit. *J. Mach. Learn. Res.* 2009; 1755–1758p.
21. H2O.ai https://www.h2o.ai/ [accessed September 2018].
22. Pedregosa Fabian, et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* 2825–2830p.
23. Pedregosa F, Weiss R, Brucher M. Scikit-Learn: Machine Learning in Python. 2011; 12: 2825–2830p.
24. scikit-learn: machine learning in Python— scikit-learn 0.19.0 documentation. Available from http://scikit-learn.org/stable/ [accessed September 2018].
25. Curtin RR, Cline JR, Slagle NP, March WB, Ram P, Mehta NA, Gray AG. MLPACK: A Scalable C++ Machine Learning Library. *Journal of Machine Learning Research* 2013; 14: 801–805p.
26. Natural Language Toolkit. *NLTK Book.* Available from www.nltk.org/ [accessed September 2018].
27. An Easy-to-Use, Easy-to-Learn Deep Learning Platform. *Paddle Paddle - Train and Deploy Deep Learning at Scale.* Available from paddlepaddle.org/ [accessed September 2018].
28. A Flexible Framework for Neural Networks. Chainer. Available from chainer.org/ [accessed September 2018].
29. Benchml-ml https://github.com/szilard/benchm-ml.
30. Rao D, Thakur D. A Review on Image Classification Approaches and Techniques. 2016; 2 (6): 356–360p.
31. Lu D, Weng Q. A Survey of Image Classification Methods and Techniques for Improving Classification Performance. *Int. J. Remote Sens.* 2007; 28 (5): 823–870p.
32. Livadas C, Walsh R, Lapsley D, Strayer WT. Using Machine Learning Techniques to Identify Botnet Traffic. *Proc. Conf. Local Comput. Networks, LCN* 2006, No. January 2014, 967–974p.
33. Tsai CF, Hsu YF, Lin CY, Lin WY. Intrusion Detection by Machine Learning: A Review. *Expert Syst. Appl.* 2009; 36 (10): 11994–12000p.
34. Sommer R, Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *2010 IEEE Symp. Secur. Priv.* 2010; 305–316p.
35. Puyalnithi T, Madhu Viswanatham V. Preliminary Cardiac Disease Risk Prediction Based on Medical and Behavioural Data Set Using Supervised Machine Learning Techniques. *Indian J. Sci. Technol.* 2016; 9 (31), 1-5p.
36. Ong JBS, Wang Z, Goh RSM, Yin XF, Xin X, Fu X. Understanding Natural Disasters as Risks in Supply Chain Management through Web Data Analysis. *Int. J. Comput. Commun. Eng.* 2015; 4 (2): 126–133p.
37. Terdik G, Gal Z. Advances and Practice in Internet of Things. *4th IEEE Int. Conf. Cogn. Infocommunications, Cog Info Com 2013 - Proc.* 2013; 435–440p.
38. Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K. Text Summarization Techniques: A Brief Survey. 2017; 8(10), 397-405p.

39. Meade F. Using Robust NLP and Machine Learning. 1997.
40. Yi J, Nasukawa T, Bunescu R, Niblack W. Sentiment Analyzer: Extracting Sentiments about a given Topic Using Natural Language Processing Techniques. *Proc. Third IEEE Int. Conf. Data Min*. 2003; 1401: 427–434p.
41. Gupta V, Lehal GS. A Survey of Text Summarization Extractive Techniques. *J. Emerg. Technol. Web Intell.* 2010; 2 (3): 258–268p.
42. Blei DM, Edu BB, Ng AY, Edu AS, Jordan MI, Edu JB. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 2003; 3; 993–1022p.
43. Blei DM, Ng AY, Jordan MI. Latent Dirichlet Allocation. *Proc. 14th Annu. Conf. Neural Inf. Process. Syst.* 2001; 601–608p.
44. Guo X, Singh S, Lewis R, Lee H. To Improve Monte Carlo Tree Search in ATARI Games. 2015.
45. Ryu S. Book Review: Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie or Die. *Healthc. Inform. Res.* 2013; 19 (1): 63p.
46. Isinkaye FO, Folajimi YO, Ojokoh BA. Recommendation Systems: Principles, Methods and Evaluation. *Egypt. Informatics J.* 2015; 16 (3): 261–273p.
47. Akansu AN, Malioutov D, Palomar DP, Jay E, Mandic DP. Introduction to the Issue on Financial Signal Processing and Machine Learning for Electronic Trading. *IEEE J. Sel. Top. Signal Process.* 2016; 10 (6): 979–981p.
48. Heaton JB, Polson NG, Witte JH. Deep Learning in Finance. 2016; No. February 1–20.
49. Amazon Machine Learning - Predictive Analytics with AWS. Available from https://aws.amazon.com/machine-learning/ [accessed September 2019].
50. Cloud Platform: Cloud Infrastructure - IBM Bluemix. Available from https://www.ibm.com/cloud-computing/bluemix/ [accessed September 2018].
51. Microsoft Azure Machine Learning Studio. Available from https://studio.azureml.net/ [accessed September 2018].
52. Machine Learning. Microsoft Azure. Available from https://azure.microsoft.com/en-in/services/machine-learning/ [accessed September 2018].
53. Chappel DA. Introducing Azure Machine Learning. *Microsoft Azur.* 2015; 17p.
54. Google Cloud Computing, Hosting Services & APIs Google Cloud Platform. Available from https://cloud.google.com/ [accessed September 2018].
55. Alibaba Cloud. An Integrated Suite of Cloud Products, Services and Solutions. Alibaba Cloud. Available from www.alibabacloud.com/ [accessed September 2018].

Cite this Article

Tushar Champaneria, Ketakee Nimavat, Maitrik Shah *et al*. A Study of Various Machine Learning and Deep Learning libraries/Platforms with Their Usage Perspective. *Journal of Artificial Intelligence Research & Advances*. 2018; 5(3): 1–15p.