UNIT-I

Fundamental Concepts of Machine Learning, Applications and Its Types

1.1 Machine Learning: A Definition

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.

1.1.1 Learning is used when:

- Human expertise does not exist (navigating on Mars)
- Humans are unable to explain their expertise (speech recognition)
- Solution changes in time (routing on a computer network)
- Solution needs to be adapted to particular cases (user biometrics)

Machine learning is programming computers to optimize a performance criterion using example data or past experience. We have a model defined up to some parameters, and learning is the execution of a computer program to optimize the parameters of the model using the training data or past experience. The model may be predictive to make predictions in the future, or descriptive to gain knowledge from data, or both.

Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM. He defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed." However, there is no universally accepted definition for machine learning. Different authors define the term differently.

1.1.2 Machine Learning: Examples

i) Handwriting recognition learning problem.

- Task T: Recognising and classifying handwritten words within images.
- Performance P: Percent of words correctly classified.
- Training experience E: A dataset of handwritten words with given classifications

ii) A robot driving learning problem:

- Task T: Driving on highways using vision sensors.

• Performance measure P: Average distance traveled before an error.

•Training experience E: A sequence of images and steering commands recorded while observing a human driver

iii) A chess learning problem:

• Task T: Playing chess.

• Performance measure P: Percent of games won against opponents.

• Training experience E: Playing practice games against itself

Definition:

A computer program which learns from experience is called a machine learning program or simply a learning program. Such a program is sometimes also referred to as a learner.

1.1.3 Components of Learning: Basic components of learning process The learning process, whether by a human or a machine, can be divided into four components, namely, data storage, abstraction, generalization and evaluation.

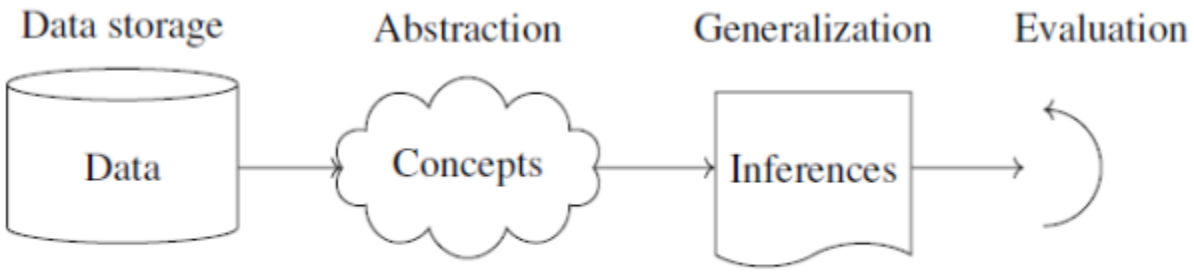


Figure 1.1: Components of learning process

Figure 1.1 illustrates the various components and the steps involved in the learning process.

1**. Data storage:** Facilities for storing and retrieving huge amounts of data are an important component of the learning process. Humans and computers alike utilize data storage as a foundation for advanced reasoning. In a human being, the data is stored in the brain and data is retrieved using electrochemical signals. Computers use hard disk drives, flash memory, random access memory and similar devices to store data and use cables and other technology to retrieve data.

2. **Abstraction**: The second component of the learning process is known as abstraction. Abstraction is the process of extracting knowledge about stored data. This involves creating general concepts about the data as a whole. The creation of knowledge involves application of known models and creation of new models. The process of fitting a model to a dataset is known as training. When the model has been trained, the data is transformed into an abstract form that summarizes the original information.

3. **Generalization**: The third component of the learning process is known as generalisation. The term generalization describes the process of turning the knowledge about stored data into a form that can be utilized for future action. These actions are to be carried out on tasks that are similar, but not identical, to those what have been seen before. In generalization, the goal is to discover those properties of the data that will be most relevant to future tasks.

4. **Evaluation:** Evaluation is the last component of the learning process. It is the process of giving feedback to the user to measure the utility of the learned knowledge. This feedback is then utilised to effect improvements in the whole learning process.

1.1.4 Areas of Influence for Machine Learning:

- *Statistics:* How best to use samples drawn from unknown probability distributions to help decide from which distribution some new sample is drawn?
- *Brain Models*: Non-linear elements with weighted inputs (Artificial Neural Networks) have been suggested as simple models of biological neurons.
- *Adaptive Control Theory*: How to deal with controlling a process having unknown parameters that must be estimated during operation?
- *Psychology*: How to model human performance on various learning tasks?
- *Artificial Intelligence*: How to write algorithms to acquire the knowledge humans are able to acquire, at least, as well as humans?
- *Evolutionary Models*: How to model certain aspects of biological evolution to improve the performance of computer programs?

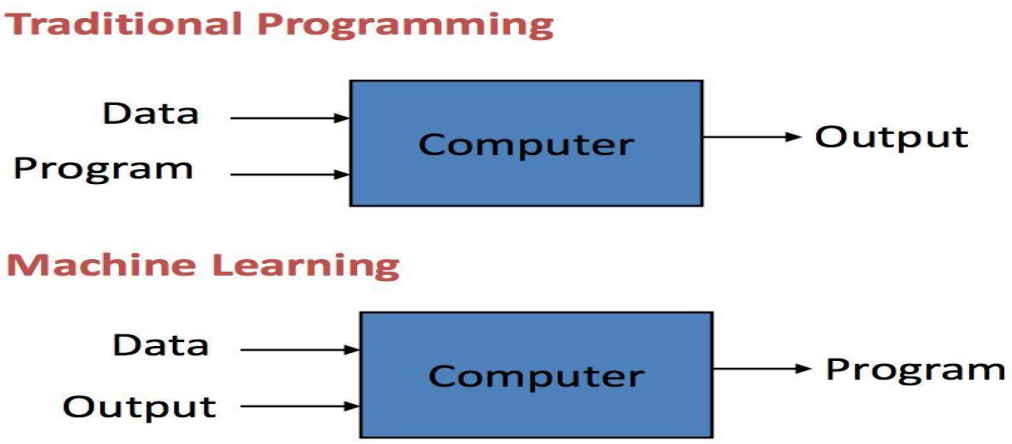
1.2 Machine Learning versus Traditional Programming

Traditional computer programming has been around for more than a century, with the first known computer program dating back to the mid 1800s. **Traditional Programming** refers to any manually created program that uses input data and runs on a computer to produce the output.

But for decades now, an advanced type of programming has revolutionized business, particularly in the areas of intelligence and embedded analytics. In **Machine Learning** programming, also

known as augmented analytics, the input data and output are fed to an algorithm to create a program. This yields powerful insights that can be used to predict future outcomes.

	Traditional Programming	**Machine Learning**
1	Traditional programming is a manual process—A person (programmer) creates the program. But without anyone programming the logic, one has to manually formulate or code rules.	In machine learning, the algorithm automatically formulates the rules from the data.
2	In traditional software, the primary objective is to meet functional and non-functional requirements.	In machine learning models, the primary goal is to optimize the metric (accuracy, precision/recall, RMSE, etc) of the models. Every 0.1 % improvement in the model metrics could result in significant business value creation.
3	The quality of the software primary depends on the quality of the code.	The quality of the model depends upon various parameters which are mainly related to the input data and hyperparameters tuning.
4	Traditional software is created using one software stack such as MEAN, Java, etc.	Machine learning models could be created using different algorithms and associated libraries. Each of these algorithms could result in different performance.
5	Feed in DATA (Input) + PROGRAM (logic), run it on the machine, and get the output.	Feed in DATA(Input) + Output, run it on the machine during training and the machine creates its own program(logic), which can be evaluated while testing.



1.3 The Seven Steps of Machine Learning.

Step 1: Data Gathering / Data Collection

Step 2: Preparing the data

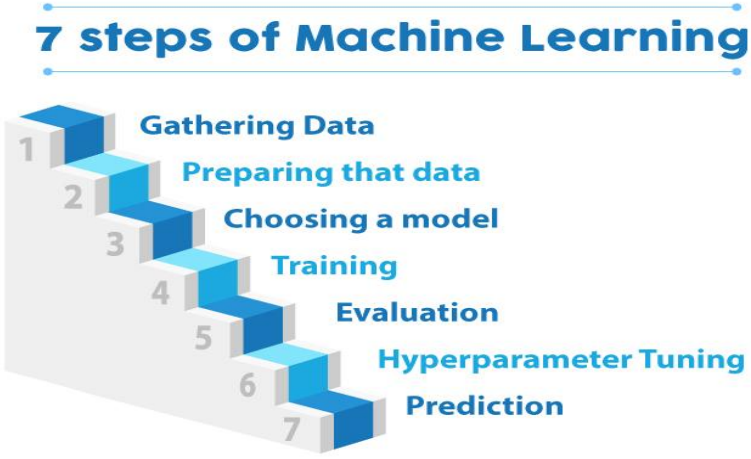
Step 3: Choosing a Model

Step 4: Training

Step 5: Evaluation

Step 6: Hyperparameter Tuning

Step 7: Prediction.



1. **Data Collection:**

It is of the utmost importance to collect reliable data so that our machine learning model can find the correct patterns. The quality of the data that you feed to the machine will determine how

accurate your model is. If you have incorrect or outdated data, you will have wrong outcomes or predictions which are not relevant.

Make sure you use data from a reliable source, as it will directly affect the outcome of your model. Good data is relevant, contains very few missing and repeated values, and has a good representation of the various subcategories/classes present.

2. **Data Preparation:**
After you have your data, you have to prepare it. You can do this by :

Putting together all the data you have and randomizing it. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process.

Cleaning the data to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. You might even have to restructure the dataset and change the rows and columns or index of rows and columns.

Visualize the data to understand how it is structured and understand the relationship between various variables and classes present.

Splitting the cleaned data into two sets, i.e., a training set and a testing set. The training set is the set from that model learns from. A testing set is used to check the accuracy of the built model after training.

3. **Choose a Model:**

   Different algorithms are for different tasks; choose the right one.

   A machine learning model determines the output you get after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand. Over the years, scientists and engineers developed various models suited for different tasks like speech recognition, image recognition, prediction, etc. Apart from this, you also have to see if your model is suited for numerical or categorical data and choose accordingly.

4. **Train the Model:**

   The goal of training is to answer a question or make a prediction correctly as often as possible

   Training is the most important step in machine learning. In training, you pass the prepared data to your machine learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

5.**Evaluate the Model:**

After training your model, you have to check to see how it's performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy.

When used on testing data, you get an accurate measure of how your model will perform and its speed.

Uses some metric or combination of metrics to "measure" objective performance of mode. Test the model against previously unseen data. This unseen data is meant to be somewhat representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not)

→ Good train/test split? 80/20, 70/30, or similar, depending on domain, data availability, dataset particulars, etc.

6. Parameter Tuning:

Once you have created and evaluated your model, see if its accuracy can be improved in any way. This is done by tuning the parameters present in your model. Parameters are the variables in the model that the programmer generally decides. At a particular value of your parameter, the accuracy will be the maximum. Parameter tuning refers to finding these values.

This step refers to hyperparameter tuning and tune the model parameters for improved performance

Simple model hyperparameters may include: number of training steps, learning rate, initialization values and distribution, etc.

7. Make Predictions:

In the end, you can use your model on unseen data to make predictions accurately.

Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model; a better approximation of how the model will perform in the real world

1.4 Applications of Machine Learning (Explanation)

- Social Media Features
- Product Recommendations
- Image & Speech Recognition
- Sentiment Analysis
- Self-driving cars
- Email Spam and Malware Filtering
- Stock Market Trading
- Medical Diagnosis
- Online Fraud Detection
- Automatic language translation.

1.5 Types of Learning:

In general, machine learning algorithms can be classified into three types:

- ***Supervised learning***
- ***Unsupervised learning***
- ***Reinforcement learning***

1.5.1 **Supervised learning**: A training set of examples with the correct responses (targets) is provided and, based on this training set, the algorithm generalises to respond correctly to all possible inputs. This is also called learning from exemplars. Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. In supervised learning, each example in the training set is a pair consisting of an input object (typically a vector) and an output value. A supervised learning algorithm analyzes the training data and produces a function, which can be used for mapping new examples. In the optimal case, the function will correctly determine the class labels for unseen instances. Both classification and regression problems are supervised learning problems. A wide range of

supervised learning algorithms are available, each with its strengths and weaknesses. There is no single learning algorithm that works best on all supervised learning problems.

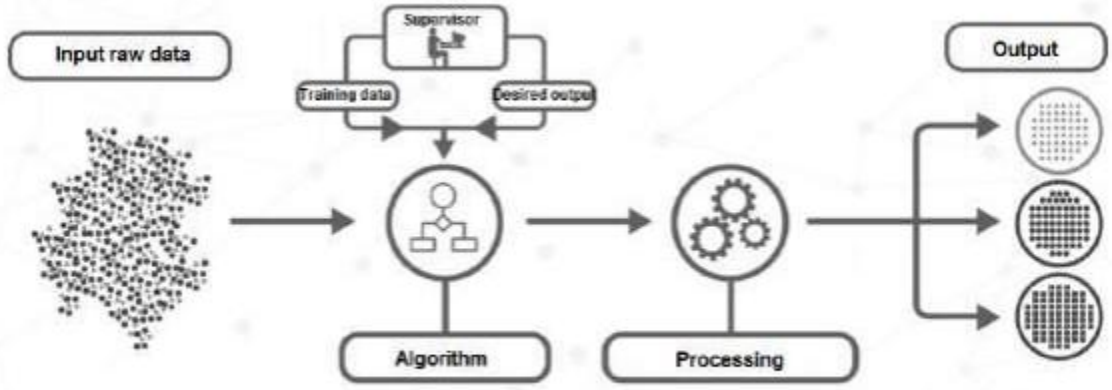


Figure 1.4: Supervised learning

A “supervised learning” is so called because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers (that is, the correct outputs), the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance. Example Consider the following data regarding patients entering a clinic. The data consists of the gender and age of the patients and each patient is labeled as “healthy” or “sick”.

1.5.2 **Unsupervised learning**: Correct responses are not provided, but instead the algorithm tries to identify similarities between the inputs so that inputs that have something in common are categorised together. The statistical approach to unsupervised learning is known as density estimation. Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. In unsupervised learning algorithms, a classification or categorization is not included in the observations. There are no output values and so there is no estimation of functions.

gender	age	label
M	48	sick
M	67	sick
F	53	healthy
M	49	healthy
F	34	sick
M	21	healthy

Since the examples given to the learner are unlabeled, the accuracy of the structure that is output by the algorithm cannot be evaluated. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns 14 or grouping in data.

gender	age
M	48
M	67
F	53
M	49
F	34
M	21

Example: Consider the following data regarding patients entering a clinic.
The data consists of the gender and age of the patients. Based on this data, can we infer anything regarding the patients entering the clinic?

1.5.3 **Reinforcement learning:** This is somewhere between supervised and unsupervised learning. The algorithm gets told when the answer is wrong, but does not get told how to correct it. It has to explore and try out different possibilities until it works out how to get the answer right. Reinforcement learning is sometime called learning with a critic because of this monitor that scores the answer, but does not suggest improvements.
Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards. A learner (the program) is not told what actions to take as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situations and, through that, all subsequent rewards. Example Consider teaching a dog a new trick: we cannot tell it what to do, but we can reward/punish it if it does the right/wrong thing. It has to find out what it did that made it get the reward/punishment. We can use a similar method to train computers to do many tasks, such as playing backgammon or chess,

scheduling jobs, and controlling robot limbs. Reinforcement learning is different from supervised learning. Supervised learning is learning from examples provided by a knowledgeable expert.

1.6 Advantages and Disadvantages of Machine Learning:

Advantages of Machine learning

1. Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

A machine can learn more when it gets more data and since it gets more data it also learns the pattern and trend for example for a social networking site like Facebook people surf and browses several data and their interest is recorded and understand the pattern and shows the same or similar trend to them to keep their interest within the same app. In this way machine learning help in identifying trends and patterns.

2. No human intervention needed (automation)

In machine learning, the whole process of data interpretation and analysis is done by computer. No men intervention is required for the prediction or interpretation of data. The whole process of machine learning is machine starts learning and predicting the algorithm or program to give the best result. One of the examples in the Google home that detect the voice and them accordingly finds out the result that the user wants, and antivirus software detects the virus of the computer and fixes it. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. As humans after gaining experience improve themselves in the same way machine learning improve themselves and become more accurate and efficient in work. This led to better decisions. For example, in the weather forecast, the more data. And experience the machine gets the more advanced forecast it will provide.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

Machine learning is used in various fields of life like education, medicine, engineering, etc. From a very small application to very big and complicated structured machines that help in the prediction and analysis of data. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning:

With all those advantages to its powerfulness and popularity, Machine Learning isn't perfect. The following factors serve to limit it:

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

There can be times when the learning process of the machine may take a lot of time because the effectiveness and efficiency can only come through experience which again requires time. ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. Sometimes data without any error could also be interpreted inaccurately by the machine as the data provided previously may not fulfil all the basics of the machine. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it. Although machine learning is considered to be more accurate it is highly vulnerable.

1.7 Most Popular Machine Learning Software Tools.

Tool	Platform	Written in language	Algorithms or Features
Scikit Learn	Linux, Mac OS, Windows	Python, Cython, C, C++	Classification Regression Clustering Preprocessing Model Selection Dimensionality reduction.
PyTorch	Linux, Mac OS, Windows	Python, C++, CUDA	Autograd Module Optim Module nn Module
TensorFlow	Linux, Mac OS, Windows	Python, C++, CUDA	Provides a library for dataflow programming.
Weka	Linux, Mac OS, Windows	Java	Data preparation Classification Regression Clustering Visualization Association rules mining
KNIME	Linux, Mac OS, Windows	Java	Can work with large data volume. Supports text mining & image mining through plugins

Colab	Cloud Service		Supports libraries of PyTorch, Keras, TensorFlow, and OpenCV
Apache Mahout	Cross-platform	Java Scala	Preprocessors Regression Clustering Recommenders Distributed Linear Algebra.
Shogun	Windows Linux UNIX Mac OS	C++	Regression Classification Clustering Support vector machines. Dimensionality reduction Online learning etc.
Keras.io	Cross-platform	Python	API for neural networks
Rapid Miner	Cross-platform	Java	Data loading & Transformation Data preprocessing & visualization.
Amazon Machine Learning (AML)	Cloud Based		It enables the user to retrieve predictions with the help of batch APIs for bulk requests or real-time APIs for individual requests.

1) Scikit-learn

Scikit-learn is for machine learning development in python. It provides a library for the Python programming language. Key concepts and features include: Algorithmic decision-making methods, including: Classification: identifying and categorizing data based on patterns.

Features:

- It helps in data mining and data analysis.

- It provides models and algorithms for Classification, Regression, Clustering, Dimensional reduction, Model selection, and Pre-processing.

Pros:

- Easily understandable documentation is provided.
- Parameters for any specific algorithm can be changed while calling objects.

Tool Cost/Plan Details: Free.

Official Website: **scikit-learn**

2) PyTorch

PyTorch is a Torch based, Python machine learning library. PyTorch is an open-source machine learning framework, which is based on **the Torch** library. This framework is free and open-source and developed by **FAIR(Facebook's AI Research lab**). It is one of the popular ML frameworks, which can be used for various applications, including computer vision and natural language processing. PyTorch has Python and C++ interfaces; however, the Python interface is more interactive. Different deep learning software is made up on top of PyTorch, such as PyTorch Lightning, Hugging Face's Transformers, Tesla autopilot, etc.**Features:**

- It enables the developers to create neural networks using Autograde Module.
- It is more suitable for deep learning researches with good speed and flexibility.
- It can also be used on cloud platforms.
- It includes tutorial courses, various tools, and libraries.
- It also provides a dynamic computational graph that makes this library more popular.
- It allows changing the network behaviour randomly without any lag.
- It is easy to use due to its hybrid front-end.
-

Pros:

- It helps in creating computational graphs.
- Ease of use because of the hybrid front-end.

Tool Cost/Plan Details: Free

Official Website: **Pytorch**

3) TensorFlow

TensorFlow provides a JavaScript library that helps in machine learning. APIs will help you to build and train the models. TensorFlow is one of the most popular open-source libraries used to train and build both machine learning and deep learning models. It provides a JS library and was developed by **Google Brain Team.** It is much popular among machine learning enthusiasts, and they use it for building different ML applications. It offers a powerful library, tools, and resources for numerical computation, specifically for large scale machine learning and deep learning projects. It enables data scientists/ML developers to build and deploy machine learning applications efficiently. For training and building the ML models, TensorFlow provides a high-level Keras API, which lets users easily start with TensorFlow and machine learning.

Features:

- TensorFlow enables us to build and train our ML models easily.
- It also enables you to run the existing models using **the TensorFlow.js**
- It provides multiple abstraction levels that allow the user to select the correct resource as per the requirement.
- It helps in building a neural network.
- Provides support of distributed computing.
- While building a model, for more need of flexibility, it provides eager execution that enables immediate iteration and intuitive debugging.
- This is open-source software and highly flexible.
- It also enables the developers to perform numerical computations using data flow graphs.
- Run-on GPUs and CPUs, and also on various mobile computing platforms.
- It provides a functionality of auto diff (Automatically computing gradients is called automatic differentiation or auto diff).
- It enables to easily deploy and training the model in the cloud.
- It can be used in two ways, i.e., by installing through NPM or by script tags.

Cons:

- It is difficult to learn.

Tool Cost/Plan Details: Free
Official Website: **Tensorflow**

4) Weka

These machine learning algorithms help in data mining.

Features:

- Data preparation
- Classification
- Regression
- Clustering
- Visualization and
- Association rules mining.

Pros:

- Provides online courses for training.
- Easy to understand algorithms.
- It is good for students as well.

Cons:

- Not much documentation and online support are available.

Tool Cost/Plan Details: Free
Official Website: **Waikato-weka**

5) KNIME

KNIME is a tool for data analytics, reporting and integration platform. Using the data pipelining concept, it combines different components for machine learning and data mining.

Features:

- It can integrate the code of programming languages like C, C++, R, Python, Java, JavaScript etc.
- It can be used for business intelligence, financial data analysis, and CRM.

Pros:

- It can work as a SAS alternative.
- It is easy to deploy and install.
- Easy to learn.

Cons:

- Difficult to build complicated models.
- Limited visualization and exporting capabilities.

Tool Cost/Plan Details: Free

Official website: **KNIME**

6) Colab

Google Colab is a cloud service which supports Python. It will help you in building the machine learning applications using the libraries of PyTorch, Keras, TensorFlow, and OpenCV

Features:

- It helps in machine learning education.
- Assists in machine learning research.

Pros:

- You can use it from your google drive.

Tool Cost/Plan Details: Free

Official Website: **Colab**

7) Apache Mahout

Apache Mahout is an open-source project of Apache Software Foundation, which is used for developing machine learning applications mainly focused on Linear Algebra. It is a distributed linear algebra framework and mathematically expressive Scala DSL, which enable the

developers to promptly implement their own algorithms. It also provides Java/Scala libraries to perform Mathematical operations mainly based on linear algebra and statistics. Apache Mahout helps mathematicians, statisticians, and data scientists for executing their algorithms.

Features:

-
- It enables developers to implement machine learning techniques, including recommendation, clustering, and classification.
- It is an efficient framework for implementing scalable algorithms.
- It consists of matrix and vector libraries.
- It provides support for multiple distributed backends(including Apache Spark)
- It runs on top of Apache Hadoop using the MapReduce paradigm.
- It provides algorithms for Pre-processors, Regression, Clustering, Recommenders, and Distributed Linear Algebra.
- Java libraries are included for common math operations.
- It follows Distributed linear algebra framework.

Pros:

- It works for large data sets.
- Simple
- Extensible

Cons:

- Needs more helpful documentation.
- Some algorithms are missing.

Tool Cost/Plan Details: Free

Official Website: **Mahout – Apache**

8) Accord.Net

Accord.Net provides machine learning libraries for image and audio processing. Accord.Net is .Net based Machine Learning framework, which is used for scientific computing. It is combined with audio and image processing libraries that are written in C#. This framework provides different libraries for various applications in ML, such as **Pattern Recognition, linear algebra,**

Statistical Data processing. One popular package of the Accord.Net framework is **Accord. Statistics, Accord.Math, and Accord.MachineLearning**.

Features:

It provides algorithms for:

- Numerical linear algebra.
- Numerical optimization
- Statistics
- Artificial Neural networks.
- Image, audio, & signal processing.
- It also provides support for graph plotting & visualization libraries.

Pros:

- Libraries are made available from the source code and also through executable installer & NuGet package manager.

Cons:

- It supports only. Net supported languages.

Tool Cost/Plan Details: Free

Official Website: **Accord.net**

9) Shogun

Shogun provides various algorithms and data structures for machine learning. These machine learning libraries are used for research and education. hogun is a free and open-source machine learning software library, which was created by **Gunnar Raetsch and Soeren Sonnenburg** in the year **1999**. This software library is written in C++ and supports interfaces for different languages such as Python, R, Scala, C#, Ruby, etc., using **SWIG**(Simplified Wrapper and Interface Generator). The main aim of Shogun is on different kernel-based algorithms such as Support Vector Machine (SVM), K-Means Clustering, etc., for regression and classification problems. It also provides the complete implementation of Hidden Markov Models.

Features:

- he main aim of Shogun is on different kernel-based algorithms such as Support Vector Machine (SVM), K-Means Clustering, etc., for regression and classification problems.
- It provides support for the use of pre-calculated kernels.
- It also offers to use a combined kernel using Multiple kernel Learning Functionality.
- This was initially designed for processing a huge dataset that consists of up to 10 million samples.
- It helps in implementing Hidden Markov models.
- It offers support for many languages like – Python, Octave, R, Ruby, Java, Scala, and Lua.

Pros:

- It can process large data-sets.
- Easy to use.
- Provides good customer support.
- Offers good features and functionalities.

Tool Cost/Plan Details: Free

Official Website: **Shogun**

10) Keras.io

Keras is an API for neural networks. It helps in doing quick research and is written in Python.

Features:

- It can be used for easy and fast prototyping.
- It supports convolution networks.
- It assists recurrent networks.
- It supports a combination of two networks.
- It can be run on the CPU and GPU.

Pros:

- User-friendly
- Modular
- Extensible

Cons:

- In order to use Keras, you must need TensorFlow, Theano, or CNTK.

Tool Cost/Plan Details: Free

Official Website: Keras

11) Rapid Miner

Rapid Miner provides a platform for machine learning, deep learning, data preparation, text mining, and predictive analytics. It can be used for research, education and application development.

Features:

- Through GUI, it helps in designing and implementing analytical workflows.
- It helps with data preparation.
- Result Visualization.
- Model validation and optimization.

Pros:

- Extensible through plugins.
- Easy to use.
- No programming skills are required.

Cons:

- The tool is costly.

Tool Cost/Plan Details:

It has four plans:

- Free plan
- **Small:** $2500 per year.
- **Medium:** $5000 per year.
- **Large:** $10000 per year.
-

12. Amazon Machine Learning (AML)

Amazon provides a great number of machine learning tools, and one of them is **Amazon Machine Learning** or AML. Amazon Machine Learning (AML) is a cloud-based and robust machine learning software application, which is widely used for building machine learning

models and making predictions. Moreover, it integrates data from multiple sources, including **Redshift, Amazon S3, or RDS.**

Features

Below are some top features:

- AML offers visualization tools and wizards.
- Enables the users to identify the patterns, build mathematical models, and make predictions.
- It provides support for three types of models, which are multi-class classification, binary classification, and regression.
- It permits users to import the model into or export the model out from Amazon Machine Learning.
- It also provides core concepts of machine learning, including ML models, Data sources, Evaluations, Real-time predictions and Batch predictions.
- It enables the user to retrieve predictions with the help of batch APIs for bulk requests or real-time APIs for individual requests.