International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET1229315

Machinet - System for Assisting Building of Machine Learning Model

Anika Bisht, Abhijeet Srivastav, Bandana Vishwakarma
Computers Science Department, BBDITM, Lucknow, India

Article Info
Volume 9, Issue 3

Page Number : 130-140

Publication Issue :
May-June-2022

Article History
Accepted : 04 May 2022
Published: 14 May 2022

ABSTRACT

As the technology of Machine Learning and Artificial Intelligence are growing rapidly, problems associated with them are growing too. Although all research focuses on the AutoML which is basically a process automation technique for machine learning research. We are proposing a system named Machinet which assists machine learning practitioners(students, engineers, professors, etc) to find a suitable model in a bunch of available models, for their use case which gives them higher accuracy.

Machinet is a software that facilitates the machine learning practitioners to test the accuracy of different machine learning models (models selected based on their use case) on their dataset and build a selected model (based on test results) by tweaking different parameters according to their need to increase the accuracy or usability.

Keywords : AutoML, Machine Learning, Artificial Intelligence

I. INTRODUCTION

This is the era of intelligent machines, their presence can be seen anywhere from advanced Robotics to a simple Washing Machine. The technology behind these smart machines is AI which is an active area of research. Currently, the subfield of Artificial Intelligence, namely Machine Learning is in vast use. Machine Learning helps us find patterns, trends, and forecasts based on given data. To be precise it is a program that generates a process when output is known contrary to the classical programming approach which generates output based on input and process defined. These smart programs are called models. There are varieties of models available for a single problem. To select a suitable Machine Learning model for a specific case, AutoML was introduced. The core technology behind AutoML is the neural network. In recent years neural networks have seen remarkable advancements and a sharp increase in their popularity. This achievement is due to long research into many aspects of the machine learning field ranging from the development of new learning techniques to new architectures. Machine learning research has a major problem of length, time, and difficulty leading to the emergence of a new field, named AutoML. The goal of AutoML is to decrease the human research time by increasing the machine compute time by automating the process (Fahlman & Lebiere, 1990; Hutter et al., 2011; Finn et al., 2017). This method was successful but modern studies can apply it too constrained search spaces which again depend heavily on reliant human design(Zoph & Le, 2016; Real et al., 2017; Tan et al., 2019). Other studies found some ways to constrain their search space to isolated algorithmic aspects, for example learning rules

used in backpropagation (Andrychowicz et al., 2016; Ravi & Larochelle, 2017), the data augmentation (Cubuk et al., 2019a; Park et al., 2019) or the intrinsic curiosity reward in reinforcement learning (Alet et al., 2019). But the main problem remains the same: these all methods remain hand-designed hence it was able to save compute time but has two major flaws. First, human-designed components bias the search results in favor of human-designed algorithms, reducing the innovation potential of AutoML. Innovation is also limited by having fewer options (Elsken et al., 2019b). Indeed, dominant aspects of performance are often left out (Yang et al., 2020). Second, constrained search spaces need to be carefully composed (Zoph et al., 2018; So et al., 2019; Negrinho et al., 2019), thus creating a new burden on researchers and undermining the purported objective of saving their time.

This issue was addressed by Esteban Real, Chen Liang, and Quoc V. Le, who proposed a method called AutoML-Zero, an automatic search for complete machine learning algorithms using a little restriction on form and only simple mathematical operations as building blocks. Other researchers also solved this issue (Li & Talwalkar, 2019; Elsken et al., 2019b; Negrinho et al., 2019) and (Silver et al 2017).
Although all research focuses on the AutoML which is basically a process automation technique for machine learning research. We are proposing a system named Machinet which assists machine learning practitioners(students, engineers, professors, etc) to find a suitable model in a bunch of available models, for their use case which gives them higher accuracy.
In summary, our contributions are

- Machinet Web, a lightweight, the easily deployable web app version of the Machinet;
- Machinet Desktop, an advanced feature-packed desktop version of the Machinet;

II. Related Works

AutoML[1]

Esteban Real, Chen Liang, and Quoc V. Le have discussed advancement in the field of Machine learning in multiple aspects, including model structures and learning methods. They have also discussed the effort to automate such research, known as AutoML, has also made significant progress. According to them, this progress has largely focused on the architecture of neural networks, where it has relied on sophisticated expert-designed layers as building blocks or similarly restrictive search spaces. They suggest that AutoML can go further: it is possible today to automatically discover complete machine learning algorithms just by using basic mathematical operations as building blocks. And they demonstrated this by introducing a novel framework that significantly reduces human bias through a generic search space. This can be achieved just by using basic mathematical operations as building blocks. Such AutoML will be able to discover complete machine learning algorithms.

Real Dream of Artificial Intelligence[2]

Matej Balog, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, and Daniel Tarlow have presented the idea that the real dream of artificial intelligence is to build systems that can write computer programs. Recently, there has been much interest in program-like neural network models but none of these can write programs; that is, they do not generate human-readable source code. Only very recently, Riedel et al. (2016); Bunel et al. (2016); Gaunt et al. (2016) explored the use of gradient descent to induce source code from input-output examples via differentiable interpreters, and Ling et al. (2016) explored the generation of source code from unstructured text descriptions. However, Gaunt et al. (2016) showed that differentiable interpreter-based program induction is inferior to discrete search-based techniques used by the programming languages community. To tackle this issue of program induction using machine learning techniques they proposed two main ideas: (a) learn to induce programs; that is, use a corpus of program

induction problems to learn strategies that generalize across problems, and (b) integrate neural network architectures with search-based methods rather than replace them.

MetaQNN - Meta Modeling Algorithm[3]

Bowen Baker, Otkrist Gupta, Nikhil Naik & Ramesh Raskar proposed the new technique of designing CNN architectures called MetaQNN, a metamodeling algorithm based on reinforcement learning to automatically generate high-performing CNN architectures for a given learning task. The learning agent is trained to sequentially choose CNN layers using Q Learning with an -greedy exploration strategy and experience replay. The agent explores a large but finite space of possible architectures and iteratively discovers designs with improved performance on the learning task. On image classification benchmarks, the agent-designed networks (consisting of only standard convolution, pooling, and fully-connected layers) beat existing networks designed with the same layer types and are competitive against the state-of-the-art methods that use more complex layer types. MetaQNN also outperformed the existing meta-modeling approaches for the network design tasks.

Concept of Fabric - Improvement in CNN[4]

Shreyas Saxena and Jakob Verbeek proposed the concept of fabric which selects an exponentially large number of architectures as compared to CNNs. The fabric consists of a 3D trellis that connects response maps at different layers, scales, and channels with a sparse homogeneous local connectivity pattern. The only hyperparameters of a fabric are the number of channels and layers. While individual architectures can be recovered as paths, the fabric can in addition ensemble all embedded architectures together, sharing their weights where their paths overlap. Parameters can be learned using standard methods based on backpropagation, at a cost that scales linearly in the fabric size. They presented benchmark results competitive with the state of the art for image classification on MNIST and CIFAR10, and for semantic segmentation on the Part Labels dataset.

Residual Learning Framework[5]

Thuy T.Pham proposed a residual learning framework to ease the training of networks that are substantially deeper than those used previously. He explicitly reformulated the layers as learning residual functions concerning the layer inputs, instead of learning unreferenced functions. He provided comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset, he evaluated residual nets with a depth of up to 152 layers—8 times deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves a 3.57% error on the ImageNet test set. This result won 1st place on the ILSVRC 2015 classification task. He also presented an analysis of CIFAR-10 with 100 and 1000 layers. Solely due to our extremely deep representations, he obtained a 28% relative improvement on the COCO object detection dataset.

Modified KNN Algorithm[6]

Gongde Guo, Hui Wang, David A.Bell proposed a modified version of classic KNN algorithms. The classic kNN algorithm is a non-parametric classification method, which is simple but effective in many cases. For a data record t to be classified, its k nearest neighbors are retrieved, and this forms a neighborhood of t. The majority voting among the data records in the neighborhood is usually used to decide the classification for t with or without consideration of distance-based weighting. However, to apply classic kNN we need to choose an appropriate value for k, and the success of classification is very much dependent on this value. In a sense, the kNN method is biased by k. In order for kNN to be less dependent on the choice of k, Wang proposed to look at multiple sets of nearest

neighbors rather than just one set of k-nearest neighbors. The proposed formalism is based on contextual probability, and the idea is to aggregate the support of multiple sets of nearest neighbors for various classes to give a more reliable support value, which better reveals the true class of t. However, in its basic form, the method is relatively slow and needs O(n2) to classify a new instance, though it is indeed less dependent on k and can achieve classification performance close to that of the best k.

Cluster Analysis[7]

Santi Satyaningsih discusses Cluster analysis which is a group of multivariate techniques whose primary purpose is to group objects based on the characteristics they possess. The cluster analysis is different from the factor analysis, because the cluster analysis group's objects, meanwhile in the factor analysis, are primarily concerned with grouping the variables. In additional information, factor analysis makes the grouping based on the patterns of variation (correlation) in the data whereas cluster analysis snakes grouping based on distance (proximity) (Hair, et al. 2010). The K-means algorithm gives the simple or flat condition, because it just gives a single set of clusters, with no particular organization or structure within them. But it can be used for cases with more distantly related data. So hierarchical clustering is the opposite of it. There are two approaches to hierarchical clustering; those are from the "bottom-up" which groups small clusters into larger ones, or from "top-down" splitting big clusters into small ones. They are called agglomerative and divisive clustering.

SVM(KNN)[8]

Janmenjoy Nayak, Bighnaraj Naik, and Dr. H. S. Behera discussed Support Vector Machine (SVM) which is a classification technique that can be applied to linear as well as non-linear data. It is a composite version of KNN mixed with SVM for visual category recognition and is augmented. In this algorithm, the training is done with the help of K's nearest neighbors to the unlabelled data point. First, the K-nearest data points are determined. Then, the pairwise distance between these K data points is computed. Thus we obtain a distance matrix from the calculated distances. A Kernel matrix is then designed from the obtained distance matrix. This kernel matrix is fed as input to the SVM classifier. The result obtained is the class of the unknown data point. Alternatively, one could use SVMs but time consumption is one of its drawbacks. Also, it involves the calculation of pairwise distances.

O-Cluster(Orthogonal partitioning Clustering)

B.G.Obula Reddy and Dr. Maligela Ussenaiah discuss a clustering method that combines a novel partitioning active sampling technique with an axis parallel strategy to identify continuous areas of high density in input space. This technique is known as O-Clustering. O cluster is a method that builds upon the contracting projection concept introduced by optigrid. O cluster makes two major contributions. First of all, it uses statistical tests to validate the quality of a cutting plane. Second, it can operate on a small buffer containing a random sample from the original data set. The partitions that do not have ambiguities are frozen and the data points associated with them are removed from the active buffer. O cluster operates repeatedly. It evaluates possible splitting points for all projections in a partition. it can select the best one and split the data into new partitions. The algorithm continues by searching for good cutting planes inside the newly created partitions. It can create a hierarchical tree structure that can translate the input space into rectangular regions. The process stages are:
(1) Loading the data buffer (2) computing histograms for active partitions (3) Finding the "best" splitting points for active partitions (4) Flag the ambiguous and "frozen" partitions (5) Split the active partitions (6) At last, reload buffer. O clusters can function optimally for huge datasets with large numbers of records and huge dimensionality.

Random Forest[10]

Biau and Scornet in their paper have done an in-depth review of theoretical and methodological aspects of the random forest algorithm, proposed by L. Breiman in 2001. The emphasis is placed on the mathematical forces driving the algorithm, with special attention given to the selection of parameters, the resampling mechanism, and variable importance measures. This algorithm has been extremely successful as a general-purpose classification and regression method. This success is due to its approach in which it combines several randomized decision trees and aggregates their predictions by averaging, and has shown excellent performance in settings where the number of variables is much larger than the number of observations. Also, it is so versatile that it can be even applied to large-scale problems and is easily adapted to various ad-hoc learning tasks, and returns measures of variable importance.

Classification[11]

Classification is a supervised machine learning process that maps input data into predefined groups or classes. The main condition for applying a classification technique is that all data objects should be assigned to classes and that each of the data objects should be assigned to only one class. Distance-based classification algorithms are techniques used for classifying data objects by computing the distance between the test sample and all training samples using a distance function. Distance-based algorithms though were originally proposed to deal with one type of data using distance-based measurements to determine the similarity between data objects. These algorithms were subsequently developed to enable the handling of heterogeneous data as real-world data sets are often diverse in types, format, content, and quality, particularly when they are gathered from different sources.

Multiple Linear Regression[12]

Regression is a statistical analysis performed to determine correlations between variables.

Univariate regression has focused on establishing a relationship between a dependent and one independent variable and formulates the linear relation equation between the dependent and independent variable. We have one more variation of regression which has one dependent and multiple independent variables called multilinear regression. Dependent and independent variables have an effect on each other or have relations and are thus used to make predictions for a subject using the relation. Gulden kaya Uyanink and Nese Guler in their paper did a descriptive analysis on multilinear regression to establish the relationship between dependent and independent variables and find out the power of the relation between them if it exists. They also analyze the influence of these variables on each other or special variables have on these variables under controlled conditions. They did this analysis while assuming that multivariate regression has a normal distribution, linearity, freedom from extreme values, and having no multiple ties between independent variables. Univariate normality, skewness, and kurtosis coefficient of variables were analyzed and it was found that independent variables are highly predictable by dependent variables, and power of relation was calculated based on the dataset they used.

III. Machinet

Machinet is a software that facilitates the machine learning practitioners to test the accuracy of different machine learning models (models selected based on their use case) on their dataset and build a selected model (based on test results) by tweaking different parameters according to their need to increase the accuracy or usability. This procedure can be a full fetch extensive(on the complete dataset) type which in end returns a complete customized machine learning model or a limited(on folded dataset) type which in

end returns a comparative study of different machine learning model performance on the test data and based on this study a model can be selected as well as customized after that an extensive model building procedure may be initiated.

Machinet comes equipped with varieties of functionalities which makes it a one-stop solution. Some of the major features of machinet are as follows:

- Helps in analyzing the performance of a specific model relative to change of the several required parameters.
- Also provides the capability to comparatively study the performance of several models on smaller parts of the test data, which helps in the selection of the best model for the specific usage.
- Generates beautiful visualizations based on user preferences.
- Wide varieties of visualizations are available like Bar, chart, heatmap, etc.
- Saves and restores trained machine learning models in a file so that they can be used to compare the model to other models or to test the model on new data. Serialization is the process of saving data, whereas Deserialization is the process of restoring it.
- Guides at each step while building the machine learning models
- Boiler Code for a selected model can be saved to be used later in the other projects.
- Easier to use, thanks to the excellent learning curve.
- And many more.

To build the machinet a lot of libraries and modules have been brought to use. Some modules have been used for ML models, others for visualization and miscellaneous usages. Also many modules and libraries have been created specifically for the machinet.

Frameworks like Tkinter and Django have been used to create the User Interface for the machinet as well as markup language and style sheets have also been used for design.

Libraries & Modules Used	
Library/Module	Usage
Pandas	Pandas is a python library for data manipulation and analysis. It provides data structures(Data frames) and operations for manipulating the data. A few of the many inbuilt methods of pandas are groping, combining, and filtering data.
Numpy	Numpy is a numerical computing open-source library of python which is used for processing multi-dimensional arrays and matrices. It provides a high-performance multidimensional array object and tools for working with these arrays. It is basically useful for linear algebra, Fourier transform, as well as random number capabilities.
Scikit-learn	This library is the most popular ML library which contains classical ML algorithms like classification, clustering,

	and regression as well as modules for dimensionality reduction, model selection, and preprocessing.
Scipy	This library contains different modules for optimization, linear algebra, integration, and statistics. It is also very useful for image manipulation.
Tensorflow	It's an open-source library by the Google Brain team to perform numerical computations with high performance. Tensorflow involves defining and running computations involving tensors. It can train and run deep neural networks that's why it is used in the field of deep learning.
Keras	It is a high-level neural networks API capable of running on top of TensorFlow. It can run seamlessly on both CPU and GPU. Keras makes it easier to build and design a Neural Network. It allows for easy and fast prototyping.
Matplotlib	Matplotlib is a data visualization and 2-D plotting library. It can be used to embed plots into applications using various GUI toolkits like Tkinter, Pyplot module also provides a MATLAB-like interface that is just as versatile as MATLAB.
Seaborn	Seaborn is a data visualization library that is based on Matplotlib and closely integrated with the NumPy and pandas data structures. Seaborn operates on data frames that have a whole dataset and then it internally performs required statistical aggregation and mapping to produce beautiful plots. It has various tools for choosing color palettes that can reveal patterns in the data.
Bokeh	is a data visualization library that is used to produce detailed graphics on various datasets, whether they are large or small. Interactive plots for modern web browsers are produced using bokeh which can be used in interactive web applications, HTML documents, or JSON

	objects.
Pygal	is a data visualization library that is used to create sexy charts! (According to their website!) Pygal is similar to Plotly or Bokeh in that it creates data visualization charts that can be embedded into web pages and accessed using a web browser, a primary difference is that it can output charts in the form of SVGs or Scalable Vector Graphics. These SVGs ensure that we can observe our charts clearly without losing any of the quality even if we scale them.
WinCenter	This library is used to find the display orientation and position the windows.
ctypes	It's a foreign function library that provides C-compatible data types and allows calling functions in DLLs or shared libraries. It can be used to wrap these libraries in pure Python.
PIL	This library adds image processing capabilities to python and provides extensive file format
	support, and an efficient internal and external representation.
Time	This is a module that provides various timer-related functions.
Threading	This module provides a simple and intuitive API for spawning multiple threads in a program. This module constructs higher-level threading interfaces on top of the lower level _thread module.
OS	This module provides a way to use operating system-dependent functionality in python. It comes under the standard python utility module.

Frameworks & other Technologies Used	
Framework/Technology	Usage
Tkinter	The Tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit. This has been used to create the UI of the desktop version of machinet.
Django	is a high-level Python web framework that focuses on fast and clean

	pragmatic design. This has been used to create the UI of the web version of machinet.
HTML	The hypertext markup language or HTML is the same old markup language for documents designed to be displayed in a web browser.
CSS	is used for describing the presentation of a record written in markup language including HTML

IV. CodeBase

The codebase of Machinet is very diversified and complex due to multiple functionalities and versions. So to manage this huge source code, an elegant code base structure is needed. We have segregated the code base into two parts one for the desktop version and another one for the web version. The proposed code base structure in the beta version is having these core packages:

Models- This package is a collection of the modules containing the base code of the available machine learning models. It functions by fitting the required parameters given by the user and returning a base or advanced model based on the parameters as specified by the user.

Screens-This package is specific to the desktop version of the machinet. This package is reserved for all the User Interface-related code, including windows, widgets, styling, etc. User Interface(UI) is designed to provide excellent User Experience(UX) as it is one of the fundamental goals of machinet.

Utilities- his package contains all the miscellaneous classes and methods which are reused multiple times in machinet but are not part of the machine learning. Example-

- SetProcessDpiAwareness()
- fromRGB()
- ImageConfigurator()
- WinCenter()
- And many more

Visualization- One of the primary functions of the machinet is to return a beautiful visualization as the outcome of the applied machine learning model on the dataset and parameters provided by the user and for this, a lot of visualization methods are available ranging from simpler bar graphs to heat maps. To buckle up all these codes this package is designated. Although the code for visualization on the desktop and web version are quite different as they are completely different environments and need different libraries for the same. That's why we have two completely different packages for both versions to handle this vigorous task of visualization:

- Plotter package for the desktop version
- Visualizer package for the web version

Apps- This package is specific to the web version and is a crucial part of a Django-based application. It contains all the code of each minor web app belonging to a single major web app(machinet). Each minor web app file contains the configuration for that specific app.

Views- This package is also specific to the web version and is somewhat similar to the Screens module of the app version and contains all the web user interface-related code which is basically in CSS and HTML.

Core- This package contains modules that tune the dimensions of the actual application window and its properties.

Assets- This directory contains all the assets used in the Machinet and is common to both the versions and includes:

- Fonts
- Images
- Audio
- And many more

Miscellaneous- There are several other files that are add ons to make machinet a good software, which are:

- setup.py- This is the first file that will install the requirements and set up the machinet according to the input received from requirements.txt and config.py
- main.py- This file gets executed after the completion of setup.py, basically it imports relevant packages and initiates the actual application.
- License.md- This is the license file and for machinet currently we are going with the 'Apache Style' permissive license.
- Main.md- It contains all the non-technical information about the machinet like what is its goal, core values, functions, aim, creators and contributors, their social media handles, and many more.
- Documentation.md- It contains all the information related to technical specifications, requirements, use cases, troubleshooting, user guide, and many more.
- DataRights.md- As the name specifies this contains all the data right declarations.
- Requirements.txt- It contains a list of all the libraries, packages, and modules required for the machinet to function.
- config.py- It contains all configuration related stuff for machinet which helps it function on the different operating systems and hardware resources available in the system like GPU (capacity, cores, and model), CPU (capacity, threads, cores, and model), VRAM (capacity), RAM (capacity), etc.
- Machinet.exe- This is the exe file that launches the application which will be further used to convert machinet into an installable.

V. Applications

Machinet offers varieties of features that extremely diversify its range of applications. It has applications for students, professionals, educators, and many more. It can be used by students to accelerate their journey of learning machine learning and deep learning. It will also help professionals in selecting appropriate machine learning models for their use cases in the early stage of their project. It can even be used by educators to make their students easily understand the

underlying concepts of machine learning and deep learning without getting stuck in the complexities of underlying code.

In general, machinet can be used:

As a tool– for testing the accuracy of different machine learning models on a small chunk of a dataset, providing the boilerplate code for the best-suited model, and many more including the comparative study

As an insight provider–It can give us some insight into the dataset and output produced by different machine learning models on the same dataset.

As a guide– It will also guide the user at each step of his journey of building or testing machine learning models by giving users brief information about the parameters, models, process, measures, etc.

VI. CONCLUSION

In this paper we have discussed the problems faced by the learners and developers in the field of machine learning and deep learning, we also discussed the setbacks of the AutoML systems. To address these issues we proposed a new system called machinet which facilitates the machine learning practitioners to test the accuracy of different machine learning models on their dataset and build a selected model by tweaking different parameters according to their need to increase the accuracy or usability with ease. We looked at the technology used and the structure of this system (in the prototype stage). We have also discussed the features of this system in detail as well as the wide range of applications it has. To design this system we also analyzed different algorithms which were required to further improve the performance.

VII. REFERENCES

[1]. AutoML Zero: Esteban Real, Chen Liang, Quoc V. Le; Google AI blog, Google Research, Brain Team

[2]. Deep Coder: Learning to Write Program: Matej Balog, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, Daniel Tarlow; Published as a conference paper at ICLR 2017

[3]. Designing Neural Architecture using Reinforcement Learning: Bowen Baker, Otkrist Gupta, Nikhil Naik & Ramesh Raskar; Published as a conference paper at ICLR 2017

[4]. Convolutional Neural Fabrics: Shreyas Saxena and Jakob Verbeek; 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

[5]. KNN Model-Based Approach in Classification: Gongde Guo, Hui Wang, David A.Bell; Conference: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE - OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003

[6]. Deep Neural Network: Thuy T.Pham; U. of Technology Sydney.

[7]. Clustering: Santi Satyaningsih; International Conference on Small and Medium Enterprises Development with a Theme "Innovation and Sustainability in SME development" (ICSMED 2012)

[8]. SVM(KNN): Janmenjoy Nayak, Bighnaraj Naik, Dr. H. S. Behera; International Journal of Database Theory and Application Vol.8, No.1 (2015), pp.169-186

[9]. O-Cluster: (Orthogonal partitioning CLUSTERing): B.G.Obula Reddy1, Dr. Maligela Ussenaiah2; IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 3, Issue 1 (July-Aug. 2012), PP 01-12

[10]. A Random Forest: Biau and Scornet; Published on 19 /5/2016 as an invited paper on Springer
[11]. Classification of Heterogeneous Data Based on Data Type Impact on Similarity: Daniel Neagu & Paul Trundle; Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK
[12]. A Study on Multiple Linear Regression: Gulden kaya Uyanink and Nese Guler; (2013) Procedia—Social and Behavioral Sciences, 106, 234-240.

Cite this article as :

Anika Bisht, Abhijeet Srivastav, Bandana Vishwakarma , "Machinet - System for Assisting Building of Machine Learning Model ", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 9 Issue 3, pp. 130-140, May-June 2022. Available at doi : https://doi.org/10.32628/IJSRSET1229315
Journal URL : https://ijsrset.com/IJSRSET1229315