Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

Mote (Model Training Environment)

Aayush Kumar[1], Rohit Aggarwal[1], Shakti Singh[1], Shubham Kushwah[1], Vishali Sharma[1]
[1]HMR Institute of Technology and Management,
Delhi-110036, India

***Abstract*: Data is the lifeblood of all businesses. Data-driven decisions make the difference between leading with the competition or fall down. Machine learning is the key to unbolting the values of companies and customers data and approving decisions that keep a company ahead in the race. The conception of the Model Training Environment is to make an interactive, visual workspace to easily build, test, and iterate on a predictive analysis model. The user can deploy a machine learning model without writing a single line of coding. The testing and training for the model are done on the same dataset. One can also summarise the dataset using a correlation matrix which helps to summarize the data and tells the relationship between the features of the data in the form of a matrix. Consequently, a pickle file for the trained model can be made to allow the user to use the implemented machine learning model anytime. Anyone who has little or no knowledge of machine learning can use the system for implementing and creating machine learning models without any hassle of programming being needed.**

Keyword:- Web Interface, Model Training, Machine Learning, Supervised Learning, Machine learning algorithms.

1. INTRODUCTION

The basic idea behind Model Training Environment is to make a visual workspace where one can implement and visualize machine learning algorithms without writing a single line of code. Through this project, we want to ease the implementation of machine learning models using Machine Learning, Django, and React. The system initially consists of uploading the structured dataset which is stored in the database. The corresponding dataset is converted into a dataframe on which the operations are performed[8]. Pandas library is used for the conversion into the dataframe. Real-world data generally contains noise, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. The probability of exceptional data has increased in today's data due to its colossal size and its start for diverse sources. Data Preprocessing is the step in which the data gets transformed, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm. Data preprocessing is required for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of the machine learning model. It involves feature encoding which is basically performing transformations on the data such that it can be easily accepted as input for machine learning algorithms while keeping its meaning intact. Basically, we are applying three techniques - Scaling, one-hot encoding, and imputation. Scaling is applied to independent variables or features of data. It is done to transform the data features into a specific scale like 0-1. It standardizes features by removing the mean and scaling to unit variance.The standard score of sample x is calculated as z = (x - u) / s.Now, one hot encoding is applied for feeding categorical data to many scikit-learn estimators, notably linear models and SVMs with the standard kernels[31]. This creates a binary queue for each category and returns a special form of matrix. We apply the one-hot encoding on the categorical features to convert them into integer values. The data may also have some missing values, which are handled by the imputation transformer. The imputation strategies-mean, median, most frequent, and constant are used according to the data. A correlation matrix, which is a table showing correlation coefficients between variables, will be plotted[10]. The correlation matrix may help to know the required labels[9]. It is a way to summarize the data and helps to understand the relationship and dependence among the features of the dataset. After the data preprocessing phase comes the algorithm selection phase that involves the implementation of various machine learning algorithms both classification and regression. These algorithms are coded using sci-kit learn library. Whenever the user selects the algorithm the code for that algorithm runs on top of the user given dataset in the Django backend Then it renders the result on the web interface in the form of predicted score.

2. RELATED WORK

Machine Learning due to its vast possibilities and demanding global market has become a key of interest for innovation, research, and design. Presently, there are several online platforms that can help in the ease implementation of machine learning algorithms. But these systems have their own shortcomings and they are not fully able to utilize the architecture and resources such as lack of centralization, dependency on the operating system, and lack of user-friendly interface. In our system, we have tried to overcome the above by uploading the dataset on our database making the computation depend only on the server rather than operating system and a user interface system, In this section, we will be discussing some innovative machine learning platforms.

An online machine learning platform is developed by Google known as Google Colab. It is a free cloud service and provides a free GPU. The hassle of creating a local environment with anaconda and installing packages is eliminated by Collaboratory. Colab solely works on Google Drive and comes with a lot of Python libraries. But it has some shortcomings like one needs to install specific libraries that do not come with standard python for every session. Also, Google Drive is our source and target for storage, which makes it eat bandwidth for large datasets[5]. Colab allows one to work on the system without interruptions but not more than that. Google Storage is used with the current session, so if someone has downloaded some files and wants

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

to use it later, it must be saved before closing the session. The online workspace developed by Microsoft known as Azure Machine Learning[7] is a better option that eliminates the coding section. It uses best-in-class algorithms and a simple drag and drop interface which allows deployment in a matter of clicks and it accelerates model creation with the automated Machine Learning User Interface access built-in feature engineering, algorithm selection, and hyperparameter sweeping to develop highly accurate models[33]. It is a secure, trusted platform and designed for responsible AI. It has the support of open source frameworks and Libraries including CubeFlow, MLflow, TensorFlow, Pytorch, Python, and R.It Build machine learning models using enterprise-grade security, compliance, and virtual network support of Azure. It is an end-to-end Machine learning lifecycle Support.

3. FEATURE SELECTION AND DATA PREPROCESSING

Data preprocessing is crucial in any data mining process as it directly impacts the success rate of the project. This reduces the complexity of the data under analysis as data in the real-world is unclean. Data is said to be unclean if it is missing attribute, attribute values, contain noise or outliers, and duplicate or wrong data. The presence of any of these will degrade the quality of the results. Our data preprocessing process mainly consists of three major steps - feature extraction, scaling, and one hot encoding.

Feature selection is one of the dimensionality reduction methods. It is used to remove irrelevant or redundant features. In addition, it improves the classification accuracy. Unlike the feature extraction methods, feature selection techniques obtain a new generalized feature set from the original set. This batch of features chosen gives the best performance due to some unbiased functions and notable criteria. To make feature selection easier and to provide users with some context about the dataset we make use of a correlation matrix. A correlation matrix[4] helps users in understanding the relationship across all features in a dataset thereby providing users useful insights about the dataset. It is a way to summarize the data and helps to understand the relationship and dependence among the features of the dataset.

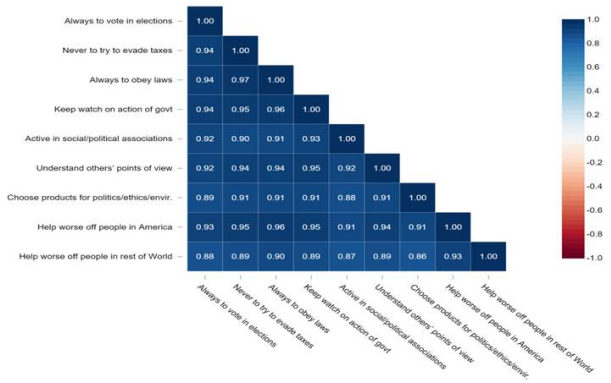

Fig.1 Sample correlation matrix plotted- Used to know the required labels

4. MODEL SELECTION

Model Selection is a vital component in machine learning modeling[15]. After the Feature Selection and Data Preprocessing phase, the Model Selection is done by the user in which the user is given choices to select the type of algorithms on the interface which involves regression and classification algorithms[18]. After choosing the generalized type the user can further choose different types of Machine learning algorithms for example, if the user chooses Regression in the start then the user can choose a different variety of regression algorithms[1] like Linear Regression, Logistic Regression, Support Vector Machine, Decision tree Regressor[2], etc. Choosing the write Machine learning algorithm is a critical point but our interface provides a flexible way to try different algorithms frequently with ease and without worrying about the code and repeating the data preprocessing steps again and again like we do when build model with the different algorithm we need to fulfill the condition according to the algorithm in the preprocessing phase[17]. The model Selection phase involves plenty of algorithms a user can try.

5. RESULTS AND FINDINGS RELATED TO OUR APPLICATION "MOTE"

From the prototype we made it is possible to implement the machine learning algorithms for any user without any prior coding knowledge. The user can implement the best possible machine learning model within a few clicks. After the model selection, a score is generated at the end which helps to understand the model selection and other phases that a user-customized earlier and further enlightens the user to improve the model by trying other machine learning algorithms[30]. The score for the machine learning model may be as high as 80 percent as the test and training on the same dataset[10].

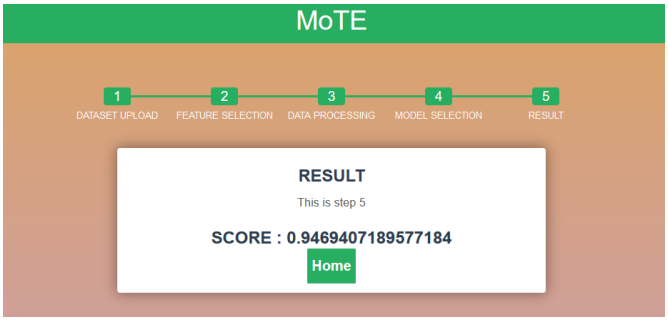

Fig - 2 Web Interface Snippet I

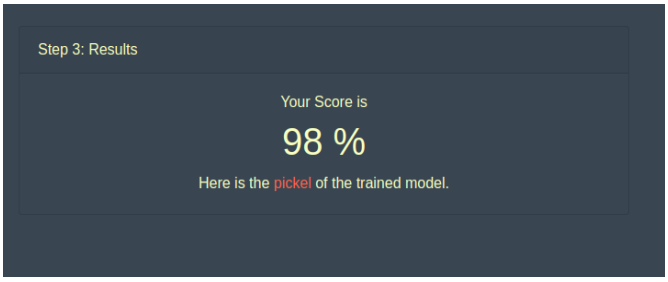

Fig. 3 Web Interface Snippet II

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

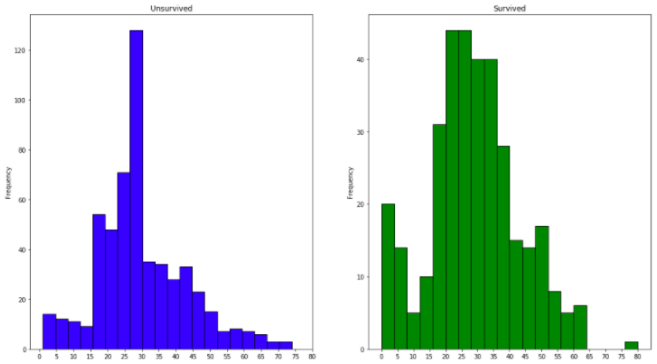

Fig. 4 Titanic survived and unsupervised data Visualization

6. CONCLUSION AND FUTURE VISION

The main idea of our application "MoTE" facilitates the user to perform all the tasks from uploading dataset to the data processing to model selection without writing a single line of code. With the help of a user-friendly interface, the application takes the dataset as input, does all the computation, and finally displays the score for the model. Generally, to implement Machine Learning, we have to have a basic understanding of a programming language (like python or R) and some packages (like NumPy, pandas, sklearn, etc) which are necessary for machine learning. Also, Machine learning models require computation power to run the algorithms. Our target is to eliminate all the mentioned hurdles in the path to start ML. Our application eliminates all the difficulties that a beginner faces if the user knows how ML works then the user can perform all the tasks and learn about the dataset more efficiently and be able to choose the best algorithms for that particular problem. It increases efficiency and conserves the time of the user. It also clears the concept of the user about the dataset. There will be no need to code from scratch if the user knows the basics of machine learning then this application can be easily used on datasets and get the required results in less amount of time.
Our future vision involves various improvements we have to do in the application which is described below:
Firstly, we have a future scope of API integration because machine learning can be daunting for the people who are not aware of it, to overcome this we can deploy our machine learning models into APIs. This can help the developers to concentrate on querying predictions by integrating Machine Learning APIs into their applications and can track events on their applications to collect usage data. Due to these factors, there is an increase in the use of machine learning APIs in application developers.

Secondly, we can have a real-time data visualization for the given dataset at the end of the result phase on the interface to get the insights of the dataset a user is dealing with so that the user can understand and analyze the dataset properly. In addition to this, our data preprocessing phase has a limited operation of scaling and normalization. So, we need to implement more customization in the data preprocessing phase which may lack in dealing with unstructured data in the future. This results in difficulty in the application of machine learning algorithms for unstructured data. Also, the addition of more advanced and improved algorithms like CNN, RNN, Xgboost, etc needs to be added in the model selection. Currently, our model doesn't support time series based modeling .so, not capable of dealing with the univariate and multivariate time series dataset. These are future scope which is to be included in the improved version of the application.

REFERENCES

[1] AK Jain, RC Dubes Algorithms for Clustering Data - 1988
[2] J Elith, JR Leathwick, T Hastie "A working guide to boosted regression trees" Journal of Animal Ecology 77 (4), 802-813
[3] Feature selection, L1 vs. L2 regularization, and rotational invariance, Andrew Y. Ng. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004
[4] Shi-Tao Yeh, GlaxoSmithKline, King of Prussia, PA "Exploratory Visualization of Correlation Matrices"
[5] Nick Mccrea" An Introduction to Machine Learning Theory And ItsApplications" A Visual Tutorial with Examples"
[6] A Very Brief Introduction to Machine Learning With Applications to Communication Systems Osvaldo Simeone, Fellow, IEEE
[7] K. P. Murphy, Machine learning: a probabilistic perspective. MIT Press, 2012.
[8] C. M. Bishop, Pattern recognition and machine learning. springer, 2006.
[9] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi, "Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks," ArXiv e-prints, Aug. 2018
[10] Johanna Hardin, Stephan Ramon Garcia and David Golan "A method for generating realistic correlation matrices", ArXiv org
[11] Liu, X. and Daniels, M. J. (2006). A new algorithm for simulating a correlation matrix based on parameter expansion and reparameterization. J. Comput. Graph. Statist. 15 897–914.
[12] T.Mitchell (1997). Machine Learning. McGraw-Hill, New York.
[13] Tu, Z. (2007). Learning generative models via discriminative approaches. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE
[14] Min-Ling Zhang, Zhi-Hua Zhou "A Review on Multi-Label Learning Algorithms", Aug. 2014, pp. 1819-1837, vol. 26
[15] I. Guyon, A. Saffari, G. Dror, and G. Cawley. Model selection: Beyond the Bayesian/Frequentist divide. JMLR, 11:61–87, 2010.
[16] D. Michie, D. Spiegelhalter, C. Taylor, and J. Campbell. Machine Learning, Neural, and Statistical Classification. Ellis Horwood, 1994
[17] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In Proc. of NIPS'12, pages 2960–2968, 2012.
[18] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In Proc. of ICML'04, page 18, 2004.
[19] G. Hamerly and C. Elkan. Learning the k in K-means. In Proc. of NIPS'04, pages 281–288, 2004.
[20] J. Vanschoren, J. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked science in machine learning. SIGKDD Explorations, 15(2):49–60, 2013.
[21] Richard Liaw Eric Liang Robert Nishihara Philipp Moritz Joseph E. Gonzalez "Tune: A Research Platform for Distributed Model Selection and Training" ICML auto workshop, ArXiv org
[22] James Bergstra, Daniel Yamins, and David Daniel Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. 2013
[23] Nicolas Gast, Stratis Ioannidis, Patrick Loiseau, and Benjamin Roussillon "Linear Regression from Strategic Data Sources", ArXiv, December 16, 2019
[24] Stratis Ioannidis and Patrick Loiseau. Linear regression as a non-cooperative game. In Proceedings of the 9th International Conference on Web and Internet Economics (WINE), pages 277–290, 2013.
[25] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, second edition, 2009.
[26] Logistic Regression Regret: What's the Catch? Gil I. Shamir, Google

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

[27] H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. The Journal of Machine Learning Research, 18(1):3117–3166, 2017
[28] Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. Journal of Machine Learning Research, 15(1):595–627, 2014.
[29] Hazan, T. Koren, and K. Y. Levy. Logistic regression: Tight bounds for stochastic and online optimization. The 27th Conference on Learning Theory, COLT 2014, page 197209. MIT Press, 2014.
[30] Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. Classification and Regression Trees. Wadsworth.
[31] Ho, T. K. 1995. Random decision forests. In Proceedings of 3rd International Conference on Document Analysis and Recognition, volume 1, 278–282. IEEE
[32] Iwasaki, M., and Miyazaki, D. 2018. Optimization of indexing based on k-nearest neighbor graphs for proximity search in high-dimensional data. arXiv preprint arXiv:1810.07355
[33] Dasgupta, S., and Freund, Y. 2008. Random projection trees and low dimensional manifolds. In STOC, volume 8, 537–546
[34] Andoni, A., and Indyk, P. 2008. Near-optimal hashing algorithms for the approximate nearest neighbor in high dimensions. Communications of the ACM 51(1