

DECISIVE ANALYSIS OF MULTIPLE LOGISTIC REGRESSION APROPOS OF HYPER-PARAMETERS

Monika Sharma

Research scholar, Dept. of Computer Science & Engineering, Poornima University
Jaipur, Rajasthan, 302022, India
Smonika15@gmail.com

Dr. Shiv Kumar Agarwal

Associate Professor, Dept. of Computer Science & Engineering, Poornima University
Jaipur, Rajasthan, 302022, India

Dr. Mahesh Bunde

Principal and Director, Poornima College of Engineering
Jaipur, Rajasthan, 302022, India

Abstract

Machine learning based predictive models are playing a vital role in every domain; however, to get the best performance of any model, hyperparameters of the underlying algorithm need to be fine-tuned. Tuning these hyperparameters is an extensive task in terms of resources and time. In this study, we tried to generalize the hyperparameters of the Multiple Logistic Regression model using the iterative method applied on multiple datasets to obtain the best hyperparameters values. In most cases, Newton-cg was found as the best solver with ridge regression. LBFGS and Liblinear were found suitable for a few datasets with L2 regularization irrespective of the linear separability of the data. This work gives the behavioral analysis of hyperparameters of the underlying dataset with respect to Multiple Logistic Regression Classification. It will help the researchers to come up with a more robust generalized tool for fine-tuning of hyperparameters to save the resources.

Keywords: Multiple Logistic Regression, Hyper-parameters, Regularization.

1. Introduction

Machine learning approaches are being applied in each domain to harness the power of vast amounts of data availability. Predictive model is one of the applications of machine learning which is being applied in a variety of domains. In predictive modelling, the model is trained using historical data and based on this learning, it is capable of predicting the target values for new data.

In our previous work, we proposed a predictive model to predict the accuracy of the Point of interest (POI) data collected through crowdsourcing [Sharma, *et al.* (2021)]. This model is based on the Multiple Logistic Regression model which was further fine-tuned to get the best performance after tweaking the hyper-parameters. From data collection to fine tuning of the model, is achieved using various stages given in Fig. 1.

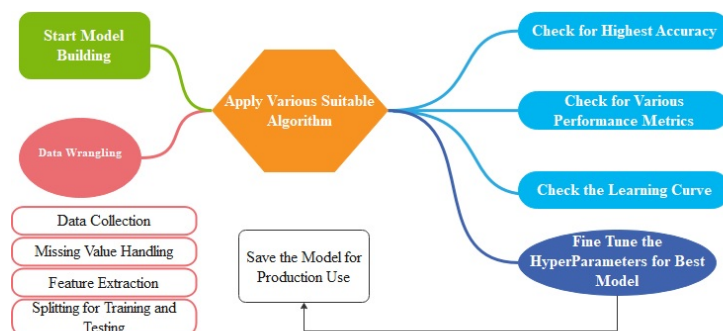


Fig. 1. This figure represents the Predictive Model Building Process

Suitable data is necessary to train the model hence data pre-processing techniques are applied to make the data appropriate for training purposes. Once the dataset is available, various machine learning algorithms are applied to get the best model. After selecting the best model, it further fine-tuned for the best values of hyper parameters to get the excellent performance of the model.

There is no perfect rule available to choose the suitable values for these hyper-parameters (HP). In this study, we tried to come up with the general guidelines which can be referred to quickly as the reference guide to choose suitable tuning parameters' values for the respective model. This study will help to develop an automated tool to provide the best possible values for HP specifically for Multiple logistic Regression (MLR). Main objectives of the paper are given below:

- (1) Analyze the relationship between HP of the model and the underlying dataset.
- (2) Developing the generic guidelines to choose the suitable values of the HP for MLR Classifier.

In this study, our focus is to analyze the HP values obtained for various domains and datasets used for MLR classification tasks. This paper is organized in five sections. Section I gives the brief idea about the purpose of the study including the model building process. Section 2, elaborate the MLR and its hyper parameters. Then we discuss the previous work done in this direction in Section 3. Section 4 is presenting the details of experimental setup and the execution outcomes. Finally, we conclude in section 5.

2. Multiple Logistic Regression

Multiple Logistic Regression (MLR) is a supervised learning algorithm used to classify the target values using the sigmoid function applied on the input feature. This sigmoid function generates the probability to make the classification decision. The MLR algorithm is mainly available for binary classification however using one v/s one and one v/s rest, it can be modified for multiclass classification problems as well. This modified version is based on the SoftMax formula for a better approximation of the expected target class. Mathematical concept of the model is based on the linear regression equation as given below:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_n x_n \quad (1)$$

Where y is the dependent variable and x_1, x_2, \dots, x_n are explanatory variables. A sigmoid function is applied to get the probability of predicted class which is given as

$$p = 1/(1 + e^{-y}) \quad (2)$$

β_0 is the intercept or bias and β_1, \dots, β_n are the coefficients which are estimated using maximum likelihood method. There is a loss function which expresses how closely the classifier produces the target label to the actual output. Our objective is to minimize the loss function or deviation in expected and the actual output. The classification outcome may suffer from overfitting if the features are perfectly fit which can increase the value of corresponding coefficient. To avoid the overfitting, regularization is added to the objective function which will penalize the large weights. There are three types of regularization used: L1 (aka Lasso), L2 (aka Ridge) and Elastic net. In case of L1 regularization, the scaled sum of the absolute values of coefficient is used whereas in case of L2, scaled sum of the square of the coefficient is used. Elastic net is the linear combination of Lasso and Ridge. A mathematical model called solver is involved to minimize the loss function with a combination of regularization. These solvers are Newton-cg, Liblinear, LBFGS, SAG and SAGA. Each one has its own merits and demerits such as Newton-cg uses the first and second order derivatives which makes it computationally expensive but works for nonlinear classification. Lib-linear is a linear classification which uses coordinate descent algorithm by successively performing approximate minimization along coordinate direction. Limited-memory Broyden-Fletcher-Goldfarb-Shanno Algorithm (LBFGS), as the name suggests, works on the limited memory by storing few vectors that represent the approximation implicitly by using an inverse Hessian matrix. Stochastic Average Gradient (SAG) optimizes the sum of smooth convex functions but it supports only L2 penalty. SAGA is a variant of SAG for non-smooth penalties. Inverse of the solver is set as C parameters. While doing the fine tuning, essentially one need to find the optimal values for all these parameters.

3. Related Work

In this paper, we analyze the HP values for the MLR algorithm used in classification work. It is important to analyze the importance of HP and the existing methods of tuning the HP.

3.1. Importance of HP tuning

Many researchers shed light on the importance of the HP tuning for performance improvement. Authors compared the performance results before fine tuning and after fine tuning. They observed the best result after applying HP tuning on various Machine learning algorithms, mainly Decision Tree, K-Nearest Neighbors and Support Vector Machine. In case of ensemble machine learning approaches, researchers found that tuning of HP greatly improved the performance of the model [Wong, *et al.* (2019)]. They proved it using a case study where a tuned model performed better. In another study, authors used 38 datasets and performed the study on six common machine learning algorithms to understand the importance of HP tuning [Probst, *et al.* (2018)]. They worked mainly on binary classification using elastic net, decision tree, KNN, SVM, random forest and XG-Boost. Researchers suggested a methodology to show the importance of HP tuning based on non-inferiority tests and tuning risk [Weerts, *et al.* (2020)]. They used 59 different data sets from open ml-cc18suite and applied SVM and random forest to study the HP tuning. In the Natural Language Processing domain also, authors found that fine tuning of HP increased the performance of the model [Rijin and Hutter (2020)]. They used 100 dataset and applied Random Forest and AdaBoost to show the importance of fine tuning of HP.

3.2. Fine tuning methods

Many researchers experimented to find the best method for fine tuning of HP. They explored grid search, random search, Bayesian optimization, particle swarm optimization and genetic algorithms on different machine learning algorithms namely Logistic Regression, Ridge classification, Support vector machine classification, Decision tree Random Forest and Naïve Bayes classification [Elgeldawi *et al.* (2021)]. Result showed the same HP values obtained using Grid, random and Bayesian for logistic regression while PSA and Genetic algorithm gives different values. They also claimed that grid, random and Bayesian gives higher accuracy. Research is going on to obtain the automated HP tuning method and so far, it is not optimized. Authors suggested the automated Machine Learning based HP optimization technique but it has a limitation to handle the changeable parameters during the training process [Wen, *et al.* (2020)]. Grid search method was found to be a more effective way for fine tuning of HP [Hossain, *et al.* (2021)]. Each method has its pros and cons such as the Grid search. All possible combinations need to be tried in case of grid search which makes it very expensive in terms of time and resources. TPOT and HPOT methods are also tried but these are suitable for regression not for classification tasks [S. Lankford (2020)].

From the literature review, it is observed that there is not much work done in obtaining an automated tool for HP tuning specifically for MLR. MLR is a basic algorithm which is a best choice for some datasets as we have seen in the case of predictive modeling of POI data verification. Therefore, we propose a detailed analysis on the values of HP which gives best results in case of MLR. Here we opted for a grid search method to obtain the best values of HP. Proposed work is significant to understand the pattern of HP values for various dataset w.r.t different statistical data properties. This work sheds the light for building an automated fine-tuning tool for MLR.

4. Experimental Setup and Execution Outcomes

To develop an automated tool for HP tuning, it is important to know the behavior of these parameters w.r.t different kinds of data sets. There are various fine-tuning methods available like manual search, Grid search, cross validation, Bayesian method ANN method etc. Manual search is the simplest process where the parameter values are provided based on the experiment. After that, analyze the obtained results and then repeat until we get the satisfactory results. In random search, a cross validation method is adopted where the training set is divided into several partitions to avoid the overfitting. Set of hyper-parameters were passed while applying the random search cross validation to get the best suitable parameters. Automated searching of the best hyper-parameters can be done using Bayesian Optimization or Genetic Algorithms. Bayesian optimization works to minimize the loss function using previous iteration's output whereas the genetic algorithm applies a natural selection method to get the best possible value. In the grid Search method, a grid is prepared using all possible combinations of the parameters. Grid search is slower than the other methods but it is more effective as all possible combinations can be tried. In this paper, we adopted the Grid search method for finding best values for the hyper-parameters for entirety. Automated methods are still under progress for Multiple logistic regression-based models. Dataset used for this study are explained in following subsections.

4.1. Datasets

As mentioned earlier, this work is the extension of the predictive model building to verify POI data collected via crowdsourcing on map portals [Sharma, *et.al* (2021)]. Prepared POI data set has four different classes with eight independent variables and it is referred here as “POI main” dataset. During the model building, Multiple Logistic Regression (MLR) was found as the best choice with the HP tuning as 100, L2 and Newton-cg for solver strength, penalty and solver choice respectively. To understand the behavior of MLR on different datasets with respect to the HP value, 15 different datasets including “POI main” were explored thoroughly as shown in Table 1. These datasets are available in UCI machine learning repository under classification category. Main characteristics of the dataset considered for proposed study are given below:

- (1) Data Set size
- (2) Binary vs. Multi class
- (3) Balance vs. Imbalance data set
- (4) Multicollinearity in independent variables

Mnemonics (used to save the space in the document), number of classes, class distribution, total number of instances and the presence of multicollinearity of all these datasets are presented in Table 1. Various domain's datasets having diversity in number of instances, number of classes, imbalance factor, multicollinearity factors were included to make the study robust.

Sr. No.	Dataset	Mnemonic	Number of Classes	Class Distribution	Number Of Instances	Multicollinearity (VIF>5)
1	Abalone	AB	29	Imbalance	4,177	Yes
2	Accelerometer	ACC	3	Balance	153,000	NO
3	Bupa	BU	2	Imbalance	345	Yes
4	Car evaluation	CA	4	Imbalance	1,728	Yes
5	Contraceptive method choice	CMC	3	Imbalance	1,473	Yes
6	Crowd sourced mapping	CPOI	6	Imbalance	10,846	Yes
7	E. Coli Genes	EC	8	Imbalance	336	Yes
8	Glass Identification	GL	7	Imbalance	214	Yes
9	Healthy Older people	HOP	4	Imbalance	26,501	Yes
10	IRIS	IR	3	Balance	150	Yes
11	Localization	LO	11	Imbalance	164,860	Yes
12	Mammograph	ME	2	Balance	830	Yes
13	POI Main	MPOI	4	Imbalance	612	NO
14	Teaching Assistant Evaluation	TA	3	Balance	150	Yes
15	Yeast	YE	10	Imbalance	1,484	Yes

Table 1. Details of all fifteen datasets considered for proposed study

4.2. Process Adopted

Multiple execution of model building and fine tuning was done for studying the impact of various factors as shown in Fig. 2. Model building steps remain the same in each iteration like analyzing the dataset, data cleaning, model building and fine tuning of the model.

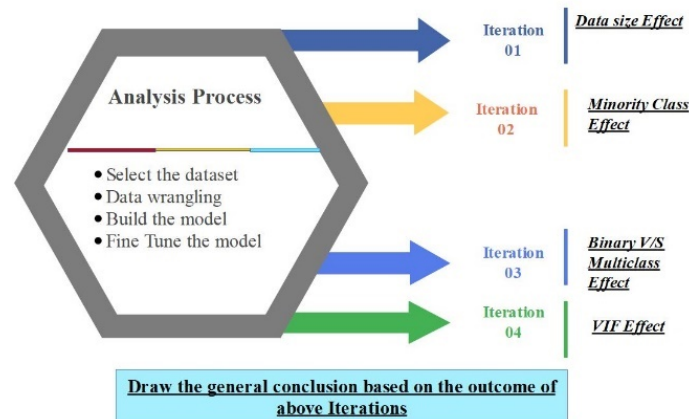


Fig. 2. This figure represents the Iterative Process applied for proposed study

Study was done in four iterations where in iteration 1, impact of dataset size was studied, imbalance versus balance class effect was studied in iteration 2, multi class v/s binary class impact on HP were studied in iteration 3 and lastly the multicollinearity relationship was analyzed in iteration 4. Observations are discussed in following section.

4.3. Execution Outcome

To check the impact of various factors, model building and HP tuning is done in various iterations and results of these iterations shown in following subsections. Experiment is done using Scikit-learn library of Python 3 on Intel(R) Core (TM) i7-7700HQ CPU @ 2.80GHz machine with 16 GB RAM.

4.3.1. Model building with original datasets

Initially the model building along with HP tuning was done for each dataset using the process explained earlier. HP values to get the best accuracy of the respective model on the dataset is given in Table 2.

Sr. No.	Dataset	Regularization	Solver	C Value
1	AB	L2	Newton-cg	1
2	ACC	L2	Liblinear	0.00001
3	BU	L2	Liblinear	0.00001
4	CA	L2	LBFGS	100
5	CMC	L2	Newton-cg	0.1
6	CPOI	None	Newton-cg	0.00001
7	EC	L2	Newton-cg	10
8	GL	L2	Newton-cg	10
9	HOP	None	Newton-cg	0.0001
10	IR	None	Newton-cg	0.0001
11	LO	L2	Liblinear	0.001
12	ME	L2	LBFGS	10
13	POI Main	L2	Newton-cg	100
14	Teaching Assistant Evaluation	L2	Liblinear	100
15	Yeast	L2	LBFGS	10

Table 2. HP values for maximum accuracy on the given dataset.

HP related to MLR, such as regularization, Solver, and C value, were considered to get the best performance. As shown in above table, for most of the datasets, Ridge regularization gives the best performance. Newton-cg and LBFGS give the best performance as a solver but for a few cases, Liblinear is better. As per the documentation of Scikit-learn, Liblinear is used in the case of linearly separable data and the Newton-cg is selected for the non-linearly separable data. But it doesn't perfectly fit with the experimental results given above, where BU and TA data are non-linearly separable. To get the in-depth relationship between data and the HP values, these datasets were tweaked for various descriptive statistical characteristics, which are presented in the following subsections.

4.3.2. Data size Impact

To check the impact of data size on the HP values, datasets were grouped according to size like BU, EC, GL, IR, TA has the instances under 500 as shown in Table 3. HP values for these datasets were not similar. The solver choice is Newton-cg for all apart from the BU and TA. These two datasets have very few classes as compared to others. The regularization scheme is also the same for most of the datasets in this group but the C value varied. The largest group is the third one where instances are between 1000 to 5000. In this group, AB and CMC show the same solver choice whereas CA and YE went with LBFGS. Regularization is the same for each case but the C value differs. CA, CMC, and YE have almost the same data size whereas AB is on the higher side. AB and YE have a large number of classes in the target.

Sr. No.	Dataset	C-Value	Solver	Regularization
#Instance less than 500	BU, EC, GL, IR, TA	0.001,10,10, 0.00001,100	Liblinear, Newton-cg, Newton-cg, Newton-cg, Liblinear	L2, L2, L2, None, L2
#Instance 500-1000	ME, MPOI	10,100	LBFGS, Newton-cg	L2 for all
#Instance 1000-5000	AB, CA, CMC, YE	1, 100, 0.1, 10	Newton-cg, LBFGS, Newton-cg, LBFGS	L2 for all
#Instance 5000-50000	CPOI, HOP	0.00001, 0.0001	Newton-cg for both	None for both
#Instance 50000-1,00,000	ACC (after slicing the data to consider only 99000)	0.00001	Newton-cg	L2
#Instance 1,00,000-2,00,000	ACC, LO	0.00001, 0.001	Liblinear	L2 for both

Table 3. Groupwise HP values for different datasets.

HP Values for Actual Size Datasets				HP Values for Reduced Size Datasets			
Dataset (Actual Size)	Regularization	Solver	C Value	Dataset (Reduced Size)	Regularization	Solver	C Value
AB1 (4177)	L2	Newton-cg	1	AB2 (1948)	L2	Newton-cg	10
ACC1(153000)	L2	Liblinear	0.00001	ACC2(99000)	L2	Newton-cg	0.00001
BU1(345)	L2	Liblinear	0.001	BU2(290)	L1	Liblinear	0.01
CM1(1473)	L2	Newton-cg	0.1	CM2(999)	L2	SAG	0.01
CSPOI1(10846)	None	Newton-cg	0.00001	CSPOI2(3000)	L2	Newton-cg	0.0001
GL1(214)	L2	Newton-cg	100	GL1(141)	L2	LBFGS	1
HOP1(26501)	None	Newton-cg	0.0001	HOP2(1946)	L2	Newton-cg	1
LO1(164860)	L1	Liblinear	0.01	LO2(44660)	L2	Liblinear	1
MPOI1(600)	L2	Newton-cg	100	MPOI2(400)	L2	Newton-cg	100
Y1(1484)	L2	LBFGS	10	Y2(648)	L2	Newton-cg	1

Table 4. Hyper-Parameters values for maximum accuracy on the given dataset (Actual vs. Reduced data size)

To understand the behavior, further we decided to divide the same dataset and obtained the HP values as shown in Table 4. The original dataset is named with suffix 1 and the reduced dataset is shown with suffix 2. It is observed that for some of the datasets, HP values were changed from the previous execution but for MPOI, no change was observed. Apart from MPOI, at least one parameter is getting changed for the altered datasets which are shown in bold in the above table. Solver value got changed for reduced dataset ACC2, GL2, and YE2 due to removal of minority classes while reducing the instances.

4.3.3. Minority Class Impact

All the classes have the same number of instances, then it is said to be a balanced class dataset. ACC, IR, ME, and TA are the balanced class datasets. HP values for these datasets are shown in Table 5.

Sr. No.	Dataset	Regularization	Solver	C-Value
1	ACC	L2	Liblinear	0.00001
2	IR	None	Newton-cg	0.00001
3	ME	L2	LBFGS	10
4	TA	L2	Liblinear	100

Table 5. HP values for the datasets with balanced classes

Ridge regularization is obtained as the best for all, except for IR but the other parameters are different in all cases. To check the effect on the same datasets, we have removed the minority classes from a few datasets and built the model to obtain the HP values for the best accuracy. Results are shown in Table 6.

(Unequal Distribution)				(Equal Distribution)			
Dataset	Regularization	Solver	C Value	Dataset	Regularization	Solver	C Value
BU1	L2	Liblinear	0.001	BU3	L1	Liblinear	0.01
CM1	L2	Newton-cg	0.1	CM3	L2	LBFGS	0.01
CSPOI1	None	Newton-cg	0.0001	CSPOI3	L2	Newton-cg	0.0001
GL1	L2	Newton-cg	100	GL3	L2	LBFGS	1
HOP1	None	Newton-cg	0.0001	HOP3	L2	Newton-cg	1
LO1	L1	Liblinear	0.01	LO3	L2	LBFGS	1
MPOI1	L2	Newton-cg	100	MPOI3	L2	Newton-cg	100
Y1	L2	LBFGS	10	Y3	L2	Newton-cg	1

Table 6. Hyper-Parameters values for maximum accuracy on the given dataset (Imbalance vs. Balance distribution)

At least one of the parameter's values got changed with the change in distribution of the classes in the target variable. It is observed that Newton-cg was the dominant solver in case of uneven distribution of classes but it got changed if the dataset was converted into a binary classification dataset after removing the minority class. Regularization choice is L2 in case of balance class provided the number of instances remains not very less as compared to the previous one. BU1 and BU3 both are binary classification datasets but vary in a number of instances. In this scenario, Regularization and solver strength got changed but the Solver choice remained the same.

4.3.3. Impact of VIF values:

Variance Inflation Factor (VIF) is the measure of multicollinearity between independent variables. Multicollinearity should not be present in the dataset to apply the MLR, hence the lesser value of VIF is desirable.

VIF range	Dataset	Regularization	Solver	C Value
VIF>10	HOP1	None	Newton-cg	0.0001
	CSPOI1	None	Newton-cg	0.0001
VIF<=10 (after features elimination)	HOP4	None	Newton-cg	0.0001
	CSPOI4	L2	Newton-cg	0.0001
	LO1	L1	Liblinear	0.01
VIF<5 (after features elimination)	ACC	L2	Newton-cg	0.0001
	CSPOI5	L2	Newton-cg	0.0001
	HOP5	L2	Newton-cg	100
	LO2	L2	Liblinear	10
	MPOI	L2	Newton-cg	100

Table 7. Grouping of datasets based on VIF value and their HP values

It is difficult to find many datasets with less VIF value. To understand its impact, we reduced the independent variables in some of the datasets and built the model again along with the HP tuning. Results are shown in Table 7. It was observed that the solver choice remained the same irrespective of the VIF value whereas regularization and C value got changed.

4.3.4. Binary vs. Multiclass:

We have only two datasets, BU and ME, having binary classification. Glass dataset was converted into a binary dataset after removing other minority classes. It is observed that as the number of classes changed in Glass dataset, solver choice got changed from Newton-cg to LBFGS as shown in Table 8.

Particular	Dataset	Regularization	Solver	C Value
Binary Classes	BU	L2	Liblinear	1
	GL3(reduced classes)	L2	LBFGS	1
	ME	L2	LBFGS	10

Table 8. HP values for binary classes datasets

The regularization scheme is the same for each case, whereas the solver choice remained the same as observed in the original data set. C value also got impacted however, these are not sufficient cases to draw a firm conclusion.

5. Conclusion

The study reveals the fact that HP values for a Machine learning model depend on the underlying dataset. Here we have considered three important Hyper-parameters of MLR such as Solver, C value, and Regularization. During the study, it was observed that Newton-cg is suitable for the datasets which have imbalance classes, not the case of binary classification, data type for independent variable is real with a wide range of values. Liblinear is mostly suitable when the number of independent variables is less as compared to number of instances and attribute values are encoded. Solver “LBFGS” is mostly suitable for binary classes or the small range of values with most of the independent variables as integers. L2 regularization is chosen as the best for less collinear dataset. This is also observed that reducing the dataset size won't change the Solver and regularization choice provided the other statistical features remains unchanged. This observation helps to save the resources in executing very huge amount of data. This study is helpful to save the efforts in fine tuning of multiclass classification using Multiple logistic regression model. However, study must be extended for more datasets including binary classification and non collinear datasets.

Acknowledgments

The authors would like to thank to UCI Irvine Machine Learning Repository for the datasets used in this study.

References

- [1] Araveeporn, A., & Kuharatanachai, C. (2019, July). Comparing Penalized Regression Analysis of Logistic Regression Model with Multicollinearity. In *Proceedings of the 2019 2nd International Conference on Mathematics and Statistics* (pp. 52-57).
- [2] Chichignoud, M., Lederer, J., & Wainwright, M. J. (2016). A practical scheme and fast algorithm to tune the lasso with optimality guarantees. *The Journal of Machine Learning Research*, 17(1), 8162-8181.
- [3] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki. (2021): Hyperparameter tuning for machine learning algorithms used for Arabic sentiment analysis, *Informatics (MDPI)*, vol. 8, no. 4, p. 79.
- [4] Gunawan, A., & Lau, H. C. (2011, January). Fine-tuning algorithm parameters using the design of experiments approach. In *International Conference on Learning and Intelligent Optimization* (pp. 278-292). Springer, Berlin, Heidelberg.
- [5] Guo, W., Yao, L., & Long, Z. (2019, August). Application of Euler Elastica Regularized Logistic Regression on Resting-state fMRI for Identification of Alzheimer's Disease. In *Proceedings of the 2019 4th International Conference on Biomedical Signal and Image Processing (ICBIP 2019)* (pp. 51-55).
- [6] H. J. P. Weerts, A. C. Mueller, and J. Vanschoren. (2020): Importance of tuning hyperparameters of machine learning algorithms, *arXiv [cs.LG]*.
- [7] Hossain, Md Riyad, and Douglas Timmer. (2021): Machine Learning Model Optimization with Hyper Parameter Tuning Approach., *Global Journal of Computer Science and Technology*
- [8] J. N. van Rijn and F. Hutter. (2018): Hyperparameter Importance Across Datasets, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- [9] J. Wong, T. Manderson, M. Abrahamowicz, D. L. Buckeridge, and R. Tamblyn. (2019): Can hyperparameter tuning improve the performance of a super learner? : A case study, *Epidemiology*, vol. 30, no. 4, pp. 521–531.
- [10] Jankovic, A., Popovski, G., Eftimov, T., & Doerr, C. (2021): The impact of hyper-parameter tuning for landscape-aware performance regression and algorithm selection. *arXiv preprint arXiv:2104.09272*.
- [11] Kaliappan Jayakumar, Srinivasan Kathiravan, Mian Qaisar Saeed, Sundararajan Karpagam, Chang Chuan-Yu, C Suganthan. (2021): Performance Evaluation of Regression Models for the Prediction of the COVID-19 Reproduction Rate, *Frontiers in Public Health*, vol.9, pp. 130-138
- [12] L. Wen, X. Ye, and L. Gao. (2020): A new automatic machine learning based hyperparameter optimization for workpiece quality prediction, *Meas. Control*, vol. 53, no. 7–8, pp. 1088–1098
- [13] Lalor, J. P., Wu, H., & Yu, H. (2017): CIFT: Crowd-informed fine-tuning to improve machine learning ability. *arXiv preprint arXiv:1702.08563*.
- [14] Lankford, S. (2020, December): Effective Tuning of Regression Models using an Evolutionary Approach: A Case Study. In *2020 3rd Artificial Intelligence and Cloud Computing Conference* (pp. 102-108).
- [15] Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., & Soatto, S. (2020): Rethinking the hyperparameters for fine-tuning. *arXiv preprint arXiv:2002.11770*.
- [16] Liu, X., & Wang, C. (2021): An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models. *arXiv preprint arXiv:2106.09204*.
- [17] Lu, Z., Chen, C., Xin, J., & Yu, Z. (2020, October): On the Auto-Tuning of Elastic-search based on Machine Learning. In *2020 International Conference on Control, Robotics and Intelligent System* (pp. 150-156).
- [18] M. Sharma, M. Bunde, V. Bothale, and M. Nawal. (2021): Fine-Tuned Predictive Model for Verifying POI Data, *International Journal of Advanced Computer Science*, vol. 12, pp. 692–704
- [19] P. Probst, B. Bischl, and A.-L. Boulesteix, (2018): Tunability: Importance of hyperparameters of machine learning algorithms, *arXiv:1802.09596*
- [20] Senaviratna, N. A. M. R., & A. Cooray, T. M. J. (2019): Diagnosing Multicollinearity of Logistic Regression Model. *Asian Journal of Probability and Statistics*, 5(2), 1-9.
- [21] Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019): Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, 17(1), 26-40.
- [22] X. Zou, Y. Hu, Z. Tian and K. Shen. (2019): Logistic Regression Model Optimization and Case Analysis, *IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 135-139

Authors Profile



Ms. Monika Sharma is currently pursuing the Ph.D. degree in Computer Science & Engineering with Poornima University, Jaipur, India. She completed M. Tech in Computer Science and Engineering in 2005. She had worked at various positions in the software industry and Academics. She was awarded the National Merit Scholarship Award from the ministry of HRD, Government of India, for excellent performance in Secondary Board Exam, Ring of Quality Award for improving Software Quality and Special recognition for enhancing the Moodle software. She has presented and published many papers in conferences and journals. She is a member of IEEE, IETE and ACM. Her area of research is GIS, Data Science, Machine Learning and Software engineering.



Dr. Shiv Kumar Agarwal did his B. Tech in 2006 in Computer Engineering, M. Tech in 2014 and Ph. D in 2020 in Computer Science & Engineering. Presently he is working as Associate Professor in Poornima University, Jaipur, India. He has more than 15 years of teaching experience in Engineering Colleges & Universities. His key areas are Data base management system, Machine Learning, Information Security and Soft Computing. His research interest includes swarm intelligence algorithms and their applications in various complex engineering design problems. He has presented and published many papers in conferences and journals.



Professor (Dr.) Mahesh M. Bunde has completed his Bachelor's degree in Electronics and Power in 1986 from Nagpur University, Master's in Electrical Power System and Doctoral in Computer Science & Engineering from Amravati University in 1990 and 2013 respectively. Presently he is working as Principal & Director, Poornima College Engineering, Jaipur, India. He is senior member of IEEE, Life member of ISTE and IEI and the member of ACM. Presently he is working as Member, STDCOM Technical & Professional Activities, and Member ExCom, IEEE Delhi Section & Secretary, IEEE Rajasthan Subsection. His areas of research interest are Machine Learning, Wearable & Pervasive Computing, Software Defined Networks, Wireless Sensor Networks, and Smart Grid Issues etc.