Automated recommending machine learning algorithms

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Computer Science Engineering, Communication and Computer Engineering

by

Prakhar Gupta- 16UCS131
Vikas Chandak - 16UCS211

Under Guidance of
Dr. Bharavi Mishra

Department of Computer Science Engineering
The LNM Institute of Information Technology, Jaipur

April 2020

The LNM Institute of Information Technology
Jaipur, India

CERTIFICATE

This is to certify that the project entitled “Automated recommending machine learning algorithms”, submitted by Prakhar Gupta(16UCS131), Vikas Chandak (16UCS211) in partial fulfillment of the requirement of degree in Bachelor of Technology (B. Tech), is a bonafide record of work carried out by them at the Department of Computer Science Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2019-2020 under my supervision and guidance and the same has not been submitted elsewhere for award of any other degree. In my/our opinion, this thesis is of standard required for the award of the degree of Bachelor of Technology (B. Tech).

Date	Adviser: Dr. Bharavi Mishra

Dedicated to Our Families and Friends

Acknowledgments

First and foremost, We would like to thank our mentor Dr. Bharavi Mishra for guiding us thoughtfully and efficiently through out this project, giving us an opportunity to work at our own pace along our own lines, while providing us with very useful directions whenever necessary.
We would also like to thank our friend Anubhav Jain for being great sources of motivation and for providing encouragement throughout the length of this project.
We offer our sincere thanks to all other persons who knowingly or unknowingly helped me complete this project .

Abstract

There are numerous steps involved in creation of a machine learning model. Essentially, computer scientists create al-gorithms that learn or ”learn” from data that contains information about the situation that you want the model to ana-lyze, with the idea being that the model will be able to perform a better analysis when it gets access to more data. Forexample, In cases like economics, you can train a model to predict market behavior . . . and in society, you can trainmodels to predict social trends.These steps can be divided into three broad categories: selecting relevant features, using the correct classifier and tuninghyperparameters. All of these steps are challenging and time consuming. So, In this study, we have created a processwhich empirically ranks and evaluates classifiers and helps the users in choosing the right methodology.This paper is an extension of the paper ”Accurate multi-criteria decision making methodology for recommending ma-chine learning algorithms” [1]. The given Methodology used human input for making decisions at key points. We haveeliminated the use of said ”experts” and made the whole process automated

Contents

Chapter *1*

Introduction

1.1 The Area of Work

Machine Learning is useful in a variety of real-life problems. The algorithm for these problems is selected manually by Data experts. The algorithm selected by the expert might be sub-optimal due to various reasons such as lack of knowledge etc. Experts usually have huge monetary costs to the business which makes the selection all the more difficult.

We assume that the data given to us is already preprocessed. This means following steps have already been performed:

- Cleaning data: Taking messy, disparate sources of information, understanding the context, and transforming the data into a single, organized table at the appropriate level of detail or aggregation, which can then be used for machine learning.
- Formulating the problem: Understanding the true business need, properly posing the right data science question, and gathering the appropriate data for the machine learning problem.

Choices in Machine Learning is a true issue in different spaces, for example, information mining business, information obtaining and thinking, research and numerous others territories . Huge business firms and exploration establishments enlist AI specialists, for example, experts, information investigators and information specialists to dissect the business information for various kinds of key arranging. For the most part, specialists pick fitting AI algorithms utilizing their heuristic information about the area furthermore, the accessible arrangement calculations. The heuristics-based algorithms determination is a hazardous task and once in a while bring about determination of a problematic presentation algorithms. The reasons may incorporate need of the total information about the area application, i.e., the datasets have distinctive characteristic attributes, and the up-and-comer classifiers have various abilities and qualities. This procedure become additional difficult when the choice of best classifier depends on numerous measures under exacting conditions and requirements.

As per the notable ”no free lunch” hypothesis, there is no classification algorithm which performs well on all kinds of use cases. Be that as it may, it tends to be made conceivable to gauge the choice of a reasonable AI calculation for an application close by. This choice procedure of the classifiers is an application subordinate errand, since it has been hypothetically and experimentally demonstrated that no AI calculation is all around predominant on all datasets because of the various qualities and highlights of the space information.

1.2 Problem Addressed

Using a process which automatically decides the best algorithm for your dataset will have huge benefits. Hence, we have created a multi criteria decision making methodology to select the best classifier. We have used ”Fuzzy Analytical Hierarchical Processing” [2] based methods to estimate relative weights for evaluation metrics. Further, classifiers are ranked based on Relative Closeness Score. The paper follows the following steps:

- A model is created for the selecting the best possible classifier from among a dataset of training examples.
- This model contains:
  - Metadata of the training datasets
  - Best classification algorithms for the training datasets
- Logistic Regression is used to predict the suitable algorithm whenever a new dataset is given.

1.3 Technology stack

Below mentioned are the technologies , frameworks , and development environment that were used for the production of the application ,also there are versions of each that were used..

1.3.1 Scikit-learn

For the Python programming language, Scikit-learn is a free programming AI library. It highlights various calculations for characterization, relapse and bunching, including linear and logistic regression, support vector machines, k-means, etc., and is intended to communicate with the Python NumPy and SciPy mathematical and logical libraries.
We used many of these algorithms to determine the best suitable algorithm for a dataset.

1.3.2 Fuzzy AHP

The analytic hierarchy process (AHP) is an organized strategy for getting sorted out and breaking down complex choices, in view of arithmetic and psychology.It speaks to an exact methodology for evaluating the loads of choice standards. Singular specialists’ encounters are used to appraise the general

sizes of variables through pair-wise comparisons.
We used Fuzzy AHP to determine the weights of various features.

1.3.3 LightGBM

LightGBM is a framework of tree based algorithms such as XGBoost, LambdaRank etc. It focuses on being fast and proficient.

Chapter *2*

Litrature Survey

Assigning Weight manually to features is a hard task by different methods such as ”Simple and Intuitive Measure (SIM)” and ”Measure Based Evaluation (MBE)”. The difference is dimensonality between evaluation and feature set is hard to determine.(Zavadskas, Zakarevicius, Antucheviciene, 2006). We can use some normalization techniques to overcome this issue.

There are plenty of other methods to evaluate classifiers. Some methods use single criteria such as accuracy or a combination of criterions such as accuracy and time and on the basis of precision, recall, F-score, sensitivity and area under the curve (AUC).Each Dataset has a different response for different machine learning algorithm. Some algorithms focuses on maximizing accuracy while ignoring other criterias (A. A. Freitas, 2006)

Adjusted ratio of ratios(ARR) is the most frequently used benchmark for algorithm evaluation. Performace of Algorithm (PAlg) which uses time, accuracy, root mean squared error among others is also used widely for algorithm evaluation and recommendation.

Chapter *3*

Proposed Work

In this section, first we characterize a bunch of general rules and afterward depict the strategy for learning algorithms evaluation on the basis of different evaluation basis.

3.1 Guidelines for algorithm Training

For reproducing this specific algorithm, a sequence of essential tasks need to be performed. To effectively play out these tasks, a bunch of rules is introduced as follows:

- **Goal and target definition**: depicts the last objective, its comparing destinations and the related requirements to accomplish the goals. For instance, the determination of ideal execution grouping calculation for multiple class issues. Here, We name objective G as ”determination of ideal grouping calculation” as objective G and ”multi-class issues” as worldwide imperative C. Accuracy, Big O Complexity etc are some relating targets against this objective.

- **Extracting internal properties of dataset**: Every Dataset has some intrinsic properties such as no. of categorical variables, amount of data etc. These properties can be taken as non target variables which can be used to predict the best algorithm.

  The best algorithm will be a combination of the target properties with the intrinsic properties. The inputs from Dataset can be used as the training data. We will build models using these classes and decide which model to use for data selection. This works when the difference in training and testing datasets have a small amount of variance. On the other hand, when the difference in training and testing datasets have high variance, the algorithm performs poorly.[5]

- Computing the best classifier for each dataset.

  – Finding out different QMM scores for each Dataset-Algorithm pair.

  – Computing Relative Closeness Score for each algorithm which will be used to rank the algorithms.

– Filling the best algorithm in the internal dataset properties.

- Each classifier has a number of parameters which are configured to control its performance on the training set. Each classifier has a unique set of parameters that can be manipulated to create a personalization that best suits the case that is being tested.

In the next sections, we have elaborated on the methodologies given above.

3.2 Extracting internal properties of dataset

Metadata is created using the following features of the dataset:Total Attributes,Total Instance Count,Total Numerical Attributes,Total Missing Count etc. We will use this metadata to differentiate between datasets.

[H]

Multiple Datasets A new Dataset containing meta properties of given datasets

$MetadataDataset$ = [ ]

every Dataset d $propertiesDataset$ = extractProperties(d)

$MetadataDataset$.append($propertiesDataset$) MetadataDataset

Procedure Of Extracting Metadata

The results for a few datasets can be seen below.

	dataset	attributes	Instance_Count	Numerical_Attributes	Missing_Count
0	Africa	14	1059	11	0
1	Breast_cai	6	569	6	0
2	Churn_Mo	14	10000	11	0
3	diabetes	9	768	9	0
4	hcc	50	165	6	0
5	heart	14	303	14	0

Example of Metadata

3.3 Selecting Best Algorithm for Training Datasets

Six classification algorithms are used to create this model : K Nearest Neighbours,Naive Bayes Classifier,Support Vector Machine,Decision Trees,Logistic Regression,Linear Discriminant Analysis.For each of the above algorithms, the following metrics are stored for each dataset : Accuracy,F-Score,Cohen Kappa Score,CPU-Training Time,CPU-testing Time.

We use Fuzzy AHP to assign weights to each of the above metrics.Normalised performance matrix is made and relative closeness is calculated .According to relative closeness, rank is calculated for each algorithm for each dataset.

[H]

Dataset d Ranking of Algorithms for that particular dataset

$RCValues$ = [ ]
every Algorithm a a.fit(d) // Training Algorithm a on Dataset d
matrix = a.scores() // Get Different scores such as Accuracy , F1-score , Testing Time etc.
RC = getRC(matrix) // Get RC value from the score
$RCValues$.append[RC , a]
rank = rankAlgorithms(RCValues)
rank
Procedure Of Selecting Algorithm

Let's look at all the functions used in the algorithm one by one.

- **fit**: A classic function used in almost all machine learning libraries.
- **scores**: This method is used to calculate different scores determined suitable for judging an algorithm. The scores include Accuracy, F-Score, CPU Training Time etc.
- **getRC**: Relative Closeness score is defined as the closeness of the algorithm to the ideal algorithm. It has been discussed extensively in the next module.
- **rankAlgorithm**: A simple sort is performed on RCScore and the algorithm with the highest RCScore is deemed suitable for the given dataset.

Algorithm	Accuracy	F-Score	Cohen Kappa Score	CPU-Training Time(100-*)	CPU-testing Time(100-*)
K Nearest Neighbours	0.877358	0.533672	0.092226614	99.998381	99.992859
Naive Bayes Classifier	0.915094	0.829977	0.662897527	99.998666	99.999329
Support Vector Machine	0.886792	0.47	0	99.951252	99.999264
Decision Trees	0.966981	0.919238	0.83848498	99.997938	99.998656
Logistic Regression	0.976415	0.940177	0.880361174	99.994865	99.99948
Linear Discriminant Analysis	0.971698	0.926846	0.853725851	99.996121	99.999387

Results for African Banking Crisis Dataset

3.4 Computing RC Score

We use combined scores of multiple evaluation matrices to create a final score which is used in the end to rank algorithms. Our idea is to test the applicant calculation and rank them as indicated by the score of their relative closeness. We are propelled by the consistency and positioning limit of TOPSIS dynamic framework with multi-rules methods. We can create RC Score by following the steps given below:

[H]

S - m*n matrix containing performance results of the given algorithm R - n*1 (single column) relative nearness score matrice

S = S_{ij} *where S_{ij} represents for evaluation metric j, value of algorithm i

r_{ij} = S_{ij}/$\sqrt{(}\sum_{i=1}^{n}$S$_{ij}^{2}$) * Normalize performance matrix

PIS_i^+ = $\sqrt{(}\sum i = 1^m$(v$_{ij} - v_j^+)^2$) *Positive Ideal Solution

NIS_i^- = $\sqrt{(}\sum i = 1^m$(v$_{ij} - v_j^-)^2$) *Negative Ideal Solution

RC = PIS_i/(PIS_i + NIS_i) for each i *Computing relative closeness with ideal algorithms

RC

Computing RC Score

We have computed RC Scores for the African banking crisis dataset. The results are shown below:

Algorithm	Accuracy	F-Score	Cohen Kappa Score	CPU-Training Time	CPU-testing Time	RC	RANK
K Nearest Neighbours	0.127013807	0.142054	0.0464178	4.287293354	4.287056606	8.889835776	5
Naive Bayes Classifier	0.132476766	0.220926	0.333637369	4.287305573	4.287333998	9.261679217	4
Support Vector Machine	0.128379547	0.125106	0	4.285272763	4.287331211	8.826089411	6
Decision Trees	0.139988335	0.244685	0.422010811	4.287274361	4.287305144	9.381263971	3
Logistic Regression	0.141354075	0.250259	0.443087165	4.28714261	4.287340472	9.409183273	1
Linear Discriminant Analysis	0.140671205	0.24671	0.429681565	4.287196459	4.287336485	9.391596154	2

Normalised performance matrix with **Relative closeness** and **Ranks** from African Banking crisis dataset

3.5 Predicting Algorithms for New Datasets

After Getting the best Algorithms, this turns into a simple Classification problem. We have used logistic regression to classify which algorithm works best on the input dataset.

Chapter *4*

Simulation and Results

4.1 Experiments

One common point in all the cons is the human intervention. So, our focus was to convert the human provided inputs to machine provided inputs. To automate this process multiple multi-class decision-making analysis is done to determine the best input that was earlier being provided by the expert. The key input provided by the experts were the weights to different metrics that were being used to determine the best classifier. To get those best set of weights we used Fuzzy AHP as our procedure to get those weights. Fuzzy AHP is one of the best procedure for multi-criteria decision-making and to get the best set of weights. The procedure to determine the best set of weights is provided in the methodologies section. This process was run multiple times on same set of datasets, till the set of weights being used were not able to give the best results for the classifier selection.

4.2 Observations

A few key Observations made are mentioned below:

4.2.1 Pros

- The algorithms that decides the best classifier, if the expert input is automated, are highly reliable and considers all the factors that can be the decision-making criteria for the best classifier selection.

- The Weka classifiers used for comparison to select the best one out of them is completely reliable and accurate.

4.2.2 Cons

- The methodology given in that research paper require human(experts) intervention at key points of the procedure.
- The algorithms developed by them to get the best classifier for dataset are completely based on the input given by the experts, which is highly ambiguous
- The ranking methodology is used by them is highly complex, which again depends on the criteria decided by the experts.

4.3 Results

35 most common multi-class learning algorithms are used for experiments which are shown in the figure Below. For convenience, we have shown only 4 rows below.

These algorithms belong to six heterogeneous families of classifiers including: probabilistic learners, functions-based learners, decision trees learners, rules-based learners, meta-learners, and miscellaneous learners. The meta-classifiers, i.e., Adaboost M1, Randomspace, and Voting are used with REPTree as the base classifier. Similarly, Dagging and Stacking are used with Naïve Bayes as the base classifier.

Results: Model vs Actual

Dataset	Classification Column	Model Prediction	Actual Result (Top 2)
Fifa	Preferred Foot	Decision Trees	1. Decision Trees 2. Linear Discriminant Analysis
Universal Bank	Credit Card	Logistic Regression	1. Naive Bayes Classifier 2. Logistic Regression
Weather Australia	Rain Today	Decision Trees	1. Decision Trees 2. Naive Bayes
Titanic	Survived	Decision Trees	1. Decision Trees 2. Linear Discriminant Analysis

4.3.1 Accuracy

To gauge precision level of the proposed technique, we calculate the correlation coefficient of spearman's rank for all datasets.

We use Average F-Score, CPU Training Time, Consistency and CPU Testing Time with equal weights for generating the recommended ranking. In the next step, we take these evaluation metrics, normalize them and then take average of their weighted sum. Then, this average is used to generate ranking for all datasets.

We get our average rank very close to 1 so we can conclusively say that our methodology is correct. It accurately ranks the algorithms and accordingly helps specialists in the choice of exact calculations under the predetermined standards. We can see the significance of Spearman's coefficient to be 0.5

which is quite significant. This is because the critical value of correlation is 0.317 and our average correlation value is far greater than this.

4.3.2 Consistency and senstivity Analysis

The choice and number or weight of the parameters influence the final recommended ranking in multi-criteria decision making [6]. Any tinkering of parameters or weights has already been shown to result in a change in the final recommended ranking [7]. Decision makers are often not in agreement with the ranks produced, so we have fully removed them [8].

The complexity of the sensitivity analysis in our case is limited to adjusting the relative weights of the parameters. We modify the weight of each criterion, i.e., Average F-score, CPU Testing Time, CPU Training Time and Consistency, one at a time, and calculate the value of the Spearman's rank correlation coefficient to see how the proposed method acts with the altered weights. The results obtained by the proposed methodology are based on the Average F-score, CPUT Testing Time, CPU Training Time and Consistency criteria with equal weights.

In the three different configurations, the performance results of the proposed approach are substantially higher than the PAlg and ARR results: all (k=35) algorithms, top k=5 algorithms, and top k=3 algorithms. For the PAlg system, a similar interpretation can be made. However this technique generates ranks on the ADA Agnostic dataset for the algorithms (with k=35), which is statistically insignificant with respect to the ideal ranking.

The performance results of the proposed method are significantly better than the results of the PAlg and ARR under the three different setup: all (k=35) algorithms, top k=5 algorithms and top k=3 algorithms. Similar interpretation can be made for PAlg method. However, this method produces ranks for the algorithms (with k=35) on the ADA Agnostic dataset, which is statistically insignificant with respect to the ideal ranking. Similarly, under all the conditions of k=35, k=5 and k=3, the findings of the ARR approach are substantially weak compared to the proposed methods.

Chapter *5*

Conclusions and Future Work

We can use techniques such as Bayesian Optimization for hyperparameter tuning of all models. Although, being more time consuming, this will ensure that we get the best model in the end. Model can be further enhanced to recommend algorithms for different kinds of datasets not only binary classification datasets.Different techniques of similarity measuring can be used instead of cosine and euclidean for best results.

Bibliography

[1] Rahman Ali et al (2006). Accurate multi-criteria decision making methodology for recommending machine learning algorithm

[2] Ying-Ming Wang, Kwai-Sang Chin (2011), Fuzzy analytic hierarchy process: A logarithmic fuzzy preference programming methodology

[3] Garcia-Cascales Lamata (2012). On rank reversal and TOPSIS method.

[4] Tzeng Huang (2011). Multiple attribute decision making: methods and applications: CRC.

[5] J.Chen et al. Robust estimation of measurement error variance/covariance from process sampling data.

[6] Leyva Lopez Carlos, 2005; Opricovic Tzeng, 2004;

[7] Thomas L. Saaty, 2006; Zavadskas, et al., 2006

[8] Goicoechea, Hansen, Duckstein, 1982; Insua French, 1991

[9] Soares, C., Brazdil, P., Costa, J. (2000). Measures to evaluate rankings of classification algorithms. In Data Analysis, Classification, and Related Methods (pp. 119-124): Springer.

[10] Andersson, A., Davidsson, P., Lindén, J. (1999). Measure-based classifier performance evaluation. Pattern Recognition Letters, 20, 1165-1173.

[11] Zavadskas, E. K., Zakarevicius, A., Antucheviciene, J. (2006). Evaluation of ranking accuracy in multi-criteria decisions. Informatica, 17, 601-618.

[12] Aha, D. W. (1992). Generalizing from case studies: A case study. In Proc. of the 9th International Conference on Machine Learning (pp. 1-10).

[13] Alexandros, K., Melanie, H. (2001). Model selection via meta-learning: a comparative study. International Journal on Artificial Intelligence Tools, 10, 525-554.

[14] Brodley, C. E. (1993). Addressing the selective superiority problem: Automatic algorithm/model class selection. In Proceedings of the Tenth International Conference on Machine Learning (pp. 17-24).

[15] Gama, J., Brazdil, P. (1995). Characterization of classification algorithms. In Progress in Artificial Intelligence (pp. 189-200): Springer.

[16] Lindner, G., Studer, R. (1999). AST: Support for algorithm selection with a CBR approach. In Principles of data mining and knowledge discovery (pp. 418-423): Springer.

[17] Smith, K. A., Woo, F., Ciesielski, V., Ibrahim, R. (2002). Matching data mining algorithm suitability to data characteristics using a self-organizing map. In Hybrid information systems (pp. 169-179): Springer.

[18] Ali, S., Smith-Miles, K. A. (2006). A meta-learning approach to automatic kernel selection for support vector machines. Neurocomputing, 70, 173-186.

[19] Brazdil, P. B., Soares, C., Da Costa, J. P. (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning, 50, 251-277.

[20] Lim, T.-S., Loh, W.-Y., Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 40, 203-228.

[21] Romero, C., Olmo, J. L., Ventura, S. (2013). A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets. In EDM (pp. 268-271).

[22] Freitas, A. A. (2006). Are we really discovering interesting knowledge from data. Expert Update (the BCS-SGAI Magazine), 9, 41-47.

[23] Song, Q., Wang, G., Wang, C. (2012). Automatic recommendation of classification algorithms based on data set characteristics. Pattern recognition, 45, 2672-2689.

[24] Reif, M., Shafait, F., Goldstein, M., Breuel, T., Dengel, A. (2014). Automatic classifier selection for non-experts. Pattern Analysis and Applications, 17, 83-96.

[25] Gayen, A. K. (1951). The frequency distribution of the product-moment correlation coefficient in random samples of any size drawn from non-normal universes. Biometrika, 38, 219-247.

Bibliography