Vaishali Shirodkar, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 10, Issue 7, (Series-VI) July 2020, pp. 29-38

RESEARCH ARTICLE OPEN ACCESS

Web Application Model to Compare and Suggest Machine Learning Algorithm

Vaishali Shirodkar*, Navjeet Pattali**, Aditya Tari**, Ankur Yadav**
**Assistant Professor, Department of Information Technology, Goa College of Engineering, Farmagudi, Ponda, Goa*
*** Department of Information Technology, Goa College of Engineering, Farmagudi, Ponda, Goa*

ABSTRACT
For a Machine Learning task, selecting a suitable algorithm to work with is one of the most important steps. To find the appropriate algorithm for the dataset, a user might conduct experiments with a variety of algorithms, another way is to consult machine learning experts. The former solution is time consuming and may not be sufficient and the latter may not be practical in several cases. The Web application developed in this project reduces the time required to carry out the manual analysis. Algorithms from three classes of ML problem domains are embedded in the application, it generates a ranked list of algorithms with their performances on the dataset supplied by the user. The web application uses HTML and CSS for the UI and Flask Framework to embed the algorithms.

Keywords- Classification, Clustering, Performance Metric, Regression

---

Date of Submission: 10-07-2020 Date of Acceptance: 26-07-2020

I. INTRODUCTION

When we begin our way in data science we are bombarded with a large variety of algorithms belonging to different problem domains. We select an algorithm for our dataset and proceed to use it and half way through the project we realise that we are not getting appropriate results. When we further dive into the algorithm we come to know that this is not what we were looking for. This issue can be solved if there was an application which could suggest appropriate algorithms based on users data and requirement. This can be done by an application running on a server with powerful CPU and GPU along with anaconda python distribution environment. This will provide the user with recommended algorithms based on his input, without need for powerful hardware on the user end. When selecting the algorithm for the project, there is a possibility that the user can miss out on relevant algorithms. In case of an application designed to carry out this task, it can have a huge variety of algorithms listed, reducing the possibility of missing out any important algorithms.

II. IMPLEMENTATION

The application has been developed using HTML, CSS and Flask Web Framework. It contains 6 pages as shown in Fig1:

1) Dataset Upload and Problem Selection
2) Dataset Display and Algorithm Execution
3) Regression Algorithms Performance and Results
4) Classification Algorithms Performance and Results
5) Query Page to proceed with Clustering Problem
6) Clustering Algorithms Performance and Results

Upon analyzing the data given by the user, our goal is to provide appropriate algorithms from a set of well known algorithms. This application is mainly suitable for domain experts who have very good experience in their own domain but might not have expertise in using ML to automate their work. Because of their expertise in domains, experts are able to efficiently process the data and deal with its cleaning and augmentation so that it is ready for analysis.

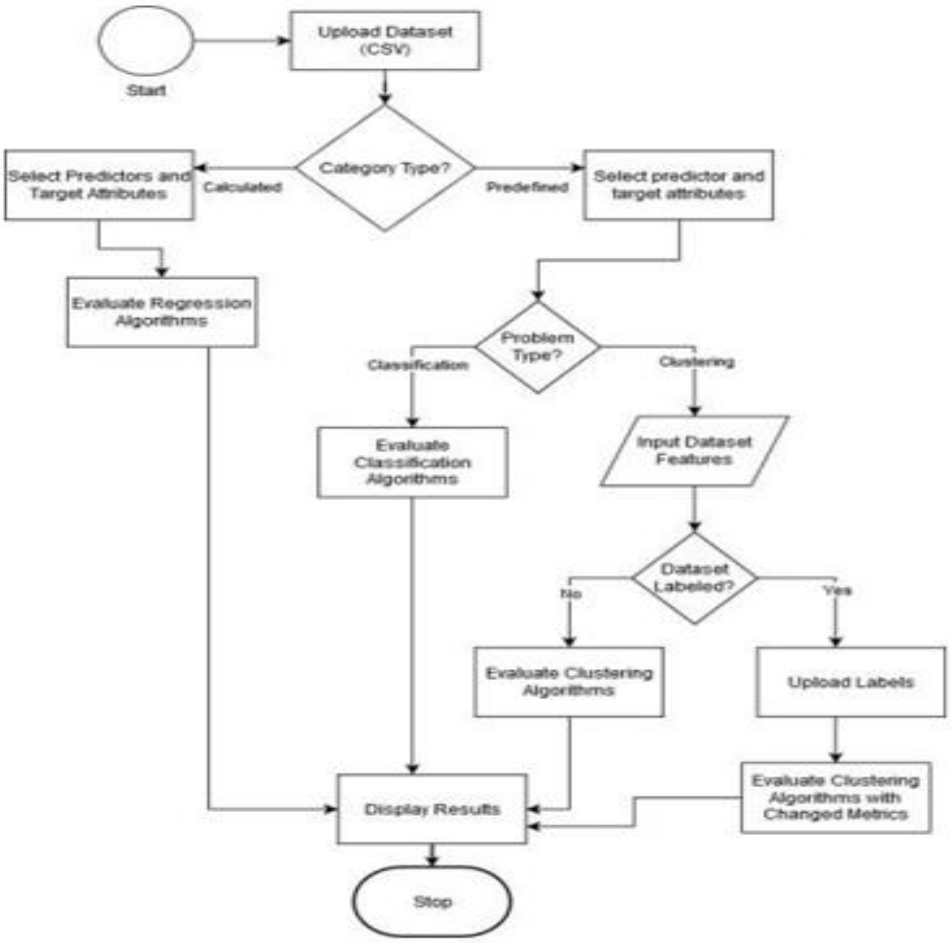

Figure 1. Application Flowchart

2.1. PROCEDURE

The user uploads the pre-processed dataset in .csv format and selects the problem domain provided below on Home Page in Figure 2. If regression is selected then the application displays the dataset head with the first 5 rows as shown in Fig 3 and the user is given the option to select the predictors and the target attributes for prediction, along with the option to execute the comparison (regression testing). The Predefined button brings the user to the common page to display the dataset and choose the predictors and the target attribute along with the option to execute classification or clustering algorithms as shown in Fig 4.

Using the provided data and the requirements, the application creates a ML model with default parameters and provides user with performance scores based on a 80-20 train test split of provided data for Regression and Classification. These models are then used to calculate and display the metric results along with ranks on individual pages. In case clustering, user is asked to input EPS and Quantile range as shown in Fig 5 according to user requirements to create models. The algorithms execute based on the information selected on this page. If the user is able to provide with the actual labeled data than he is given with accurate Adjusted Rand Index, Mutual Information based scores, Homogeneity, Completeness and V-measure based on which he is able to gauge the performance of his data using the set of popular algorithms. If the data is not labeled then it is not possible to gauge the algorithms appropriately but Silhouette Coefficient, Calinski-Harabasz Index, Davies-Bouldin Index provides user with idea about feasibility of algorithm. The plot for each algorithm is generated (in case of 2 dimensional dataset) along with the performance metric. You can find this work at https://github.com/ItProject1920/data-science-project

Figure 2. Home Page

III. EVALUATION

3.1. Algorithms

3.1.1. Regression algorithms

i) Linear Regression

Figure 3. Displaying Dataset for Regression Analysis

Figure 4. Displaying Dataset for Classification and Clustering Analysis

Figure 5. Dataset Features Input Page

ii)XGBoost
iii) Decision Tree
iv) Ridge Regression
v)Lasso Regression
vi) Elastic Net Regression
vii) Knn Regression
viii) SVM Regression

3.1.2. Classification algorithms
i) Decision Tree
ii)Random Forest
iii) Logistic Regression
iv) SVM classifier
v)Knn Classifier
vi) Ada Boost
vii) Gaussian Naive Bayes
viii) MLP classifier
ix) Multinomial Naive Bayes
x)Quadratic Discriminant Analysis

3.1.3. Clustering algorithms
i) MiniBatch Kmeans
ii)Kmeans Clustering
iii) Affinity propagation
iv) Mean Shift
v)DBSCAN
vi) OPTICS
vii) Spectral Clustering
viii) BIRCH
ix) Gaussian Mixture

3.2. Performance Metrics
3.2.1. Regression Algorithms
i) R-Square: Estimates the goodness of fit of the regression model.
ii)Mean Squared Error (MSE): shows the square of the difference of actual and predicted values and then finding its mean.

iii) Mean Absolute Error (MAE): shows the absolute value of the difference of actual and predicted values and finding the mean.
iv) Root Mean Square Error (RMSE): shows the root of the MSE value obtained.
v)Explained Variance Score: Involves finding the difference of estimated target output and the actual target output then find the ratio of the variance with reference to the difference and the variance with reference to the correct target output, finally 1 is subtracted from the ratio.

3.2.2. Classification Algorithms
i) Accuracy: Correct number of predictions made over all the predictions made
ii)Precision: Percentage of results which are relevant.
iii) Recall: Percentage of relevant results correctly classified.
iv) F1-score: Weighted average of precision and recall.
v)Confusion matrix: A table which displays performance of the model on a dataset for which true values are known.

3.2.3. Clustering Algorithms
Labeled
i) Rand Index Adjusted is a function that measures the chance of grouping elements, ignoring permutations and with chance normalization.
ii)Mutual Information based score is a function that measures the agreement of the two assignments, ignoring permutations.

iii) Homogeneity describes the closeness of the clustering algorithm to this perfection.
iv) Completeness describes the closeness of the clustering algorithm to this perfection.
v) V-measure is the computation of harmonic means of homogeneity and completeness.

Not Labeled
i) The Silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from 1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.
ii) Davies-Bouldin Index signifies the average 'similarity' between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves.
iii) Calinski-Harabasz Index is the ratio of the sum of between clusters dispersion and of inter cluster dispersion for all clusters (where dispersion is defined as the sum of distances squared).

IV. DISCUSSION

We demonstrate the working of the algorithms and their performance metrics, the comparisons are displayed on the application along with their ranks. When the user supplies the dataset (for this example we use concrete compressive strength dataset for regression and wine quality dataset for classification) the attributes selection option is displayed, with the selected problem.

In case of regression, XGBoost regression performed the best with 3 selected predictor attributes. Choosing the right attributes does affect the performance of each algorithm. The algorithms also show a change in performance based on the selected relevant attributes and the number of selected attributes. The algorithms are sorted in such a way that a lower values of R.M.S.E., M.S.E., M.A.E. and higher values of R-squared and E.V.S. are considered a better algorithm.

	Algorithm	EVS	R2	MSE	MAE	RMSE
0	XGBoost Regression	0.851374	0.850791	34.333725	4.365776	5.859489
2	Decisiontree Regression	0.773174	0.772930	52.249913	5.084563	7.228410
6	Knn Regression	0.705271	0.696473	69.843034	6.033170	8.357214
7	Support Vector Regression	0.637225	0.618958	87.679701	7.334554	9.363744
4	Lasso Regression	0.445466	0.443101	126.145157	8.692553	11.320122
5	Elastic Net Regression	0.445359	0.442996	126.169267	8.692598	11.321187
3	Ridge Regression	0.444463	0.441672	126.428028	8.714221	11.332609
1	Linear Regression	0.444462	0.441670	126.428326	8.714245	11.332622

Figure 6. Regression Result

In case of classification, the user has selected 3 predictor attributes and a target attribute to be predicted. We can see that the Random Forest algorithm is the best performer for the chosen predictors, the following

3 algorithms performances are similar and the remaining algorithms have similar values. The weighted average of the metrics has been used to calculate the rank value. The precision, recall and F1-score values displayed in the figure are the weighted average of the values for each predicted class. There is no "best algorithm" as such; the performances vary with the selected predictors, the number of predictors and also with the significance of the predictors. As seen from these examples users are able to quickly gauge the performance based on attributes and get appropriate evaluation.

	Algorithm	Accuracy	Precision	Recall	FScore	Rank
1	RandomForest	0.66875	0.653929	0.66875	0.660420	2.651848
0	Decision Tree	0.61250	0.645169	0.61250	0.626731	2.496900
8	Quadratic Discriminant Analysis	0.55625	0.542915	0.55625	0.546430	2.201845
7	Gaussian Naive Bayes	0.53125	0.474161	0.53125	0.501083	2.037744
4	Knn Classifier	0.48125	0.464176	0.48125	0.465708	1.892384
3	SVM Classifier	0.48125	0.513670	0.48125	0.385794	1.861964
2	Logistic Regression	0.48125	0.432500	0.48125	0.430048	1.825048
6	ADA Boost	0.43750	0.466632	0.43750	0.408263	1.749895
5	MLP Classifier	0.43750	0.456411	0.43750	0.293045	1.624456
9	MultinomialNB	0.43125	0.185977	0.43125	0.259880	1.308356

Figure 7. Classification Result

There are two cases in clustering problem:

i) Users heave labelled data which can be used to verify the machine clusters with actual values. In this case the user is able to get a good model with score close to 1 (eg. DBSCAN and OPTICS and if the number of clusters are already known then spectral clustering). Using this performance analysis user can accurately guess the algorithm which is feasible for his dataset.

ii) Users do not have labeled data so the analysis cannot be as accurate as incase with labeled data but silhouette scores close to zero and high Davies-Bouldin and low Calinski-Harabasz scores can be used to get a pretty accurate guess about the algorithm.

V. CONCLUSION

The Web application will provide the users with an automated service to select the most suitable algorithm with performance details for understanding. The application provides algorithms from 3 problem domains. User can upload the dataset in .csv format and the dataset should be pre-processed. The performances of the algorithms are displayed using performance metrics for each problem domain. Regression and Classification algorithms are presented in a ranked list and Clustering algorithm displays the best algorithm with a highlight and plots of clusters for visualisation of 2D data.

The project has limitations which will possibly be addressed in the future. The dataset can only be uploaded in .csv format. Regression and Classification algorithms work


Explore Machine Learning Algorithms On Your Dataset
Clustering Analysis
MiniBatchKmeans
Kmeans
Affinity Propagation
Mean Shift
DBSCAN

Figure 8. Clustering Results with Labels

Figure 9. Clustering without Labels

on default parameters, the user does not have the option to adjust them. The number of algorithms and parameters can be increased for more options. Clustering algorithm is working on default parameters for Birch and Gaussian Mixture.

REFERENCES

[1]. https://archive.ics.uci.edu/ml/datasets.php?format=task=cl aatt=area=numAtt=numIns=1 UCI Machine Learning Repo sitory: Data Sets (2018)
[2]. https://muthu.co/understanding-the-classification-report-in-sklearn/ Understanding the Classification report through sklearn - Muthukrishnan (2018)
[3]. https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation
[4]. Clustering Performance Evaluation (2019)
[5]. https://flask.palletsprojects.com/en/1.1.x/templating/#jinja-setup Flask Jinja setup (2019)
[6]. https://flask.palletsprojects.com/en/1.1.x/errorhandling/error-handlers Error handlers (2019)
[7]. https://xgboost.readthedocs.io/en/latest/ XGBoost Documentation (2020)
[8]. https://scikit-learn.org/stable/supervised_learning:html Supervised learning (2019)

Vaishali Shirodkar, et. al. "Web Application Model to Compare and Suggest Machine Learning Algorithm." *International Journal of Engineering Research and Applications (IJERA)*, vol.10 (07), 2020, pp 29-38.