Analyzing Ensemble Learning Algorithms using Cluster Analysis of Binary Classification Problems

Adiba Mahbub Proma
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
proma.adiba@gmail.com

Fahim Ishrak
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
fahim.ishrak1@gmail.com

Tahsin Mostafa
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
mostafatahsin7@gmail.com

Mahbub Alam Majumdar
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
majumdar@bracu.ac.bd

***Abstract*—With the numerous machine learning algorithms available, one of the biggest challenges while encountering a machine learning problem is finding and applying the right algorithm. Picking unsuitable algorithms result in inefficient use of resources, which can lead to massive losses for researchers and businesses. Since the No-free-lunch theorem states that there is no universal learner that works best for all problems, our goal should be to look for algorithms that are best suited for specific types of machine learning problems. Our research aims to compare three ensemble learners- Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGB)- in terms of accuracy and time complexity on different binary classification problems. The datasets are defined according to simple measures, statistical measures, and information theory measures. Similar datasets are clustered together using Self-Organizing Maps (SOM) and patterns in terms of accuracy and time complexity are found. The results show that algorithms work similarly for datasets in the same cluster with the exception of some anomalies. Finally, using the results of the algorithms from the clusters, we propose a scoring method to rank the algorithms in terms of accuracy and complexity. This method is then defined mathematically, to give a generalised rule to find what is the best suited algorithm for a specific problem according to the users requirement(proportion of weight given to accuracy, time complexity, and other performance metrics). Hence, through our research, we can predict which algorithm would work best for a particular dataset by first fitting the dataset into one of the clusters, then running the algorithms and then by using the rules generated to rank the algorithms accordingly.**

***Index Terms*—Computational learning theory, Ensemble learning, Binary classification, Multivariate data analysis, Hierarchical Agglomerative Clustering, Rule generation**

I. Introduction

The world of data analytics has been revolutionized with the various machine learning techniques available. Machine learning algorithms, which often differ in terms of learnability, time complexity, ease of use and so on, are now able to solve a variety of classification problems such as credit risk assessment, quality control in manufacturing, disease detection, object classification and so on [1]. However, the real challenge for analysts now is finding out the right algorithm suited for a specific problem. Picking unsuitable algorithms results in inefficient data analysis leading to financial loss, economic damage, and wastage of time, energy and resources. As the No-free-lunch theorem states, there is no universal learner that works best for all problems. [2] However, it can be expected that an algorithm would work well for similar datasets. One way of finding similar datasets would be by using clustering techniques to cluster the datasets. In order to do so, one has to come up with the right set of clustering metrics.These clusters can then be analysed in terms of various performance metrics as different algorithms are tested on them.

Our research aims find out what sort of problems ensemble machine learning algorithms work best for. The data sets are described using simple, statistical, and information theory measures. Using these descriptions, datasets are then clustered using Self-Organizing Maps. Then, Random Forest, Gradient Boosting and Extreme Gradient Boosting are trained on the datasets and their performance is analyzed in terms of negative log loss and training time complexity. By conducting this research, we aim to solve the problem of picking the right machine learning algorithm by mapping a hard, computationally-exhaustive problem to an easier, less expensive one.

The remaining sections of the paper has been organized as follows. Section 2 presents the literature review for our research. Section 3 gives a brief background analysis of the theory of algorithms, and Section 4 describes the metrics we will use to define the dataset. Section 5 provides the research methodology used; section 6 portrays the experimental results and section 7 explains pattern analysis of the results. Section 8 introduces the rule generation along with an example. Finally, chapter 9 deals with conclusion and further work that can be done.

II. Literature Review

One of the most comprehensive projects to compare machine learning algorithms is the STATLOG project which uses 12 different datasets, which are first defined using statistical

measures such as canonical discriminant correlation, univariate skewness, kurtosis, and so on. [3]. Both objective measures and subjective measures are used as performance metrics to compare the results. One of the biggest limitations of this paper is that it uses univariate skewness and kurtosis to define the datasets, which fails to capture the relationship of skewness and kurtosis between the features. Thus, we can infer that using univariate analysis for multivariate data can lead to invalid results since the nature of multivariate data is different from univariate data.

Taking the works of the Statlog project further, [4] compares 33 different algorithms on 16 data sets. The study also analyses the effect of noise on datasets. In more recent years, [5] compares algorithms using 100 classification datasets, which makes the result of the project more reliable. The metrics used to describe the data sets are simple measures, statistical measures and information theory.Finally, the paper generates rules using the rule-based algorithm C5.0 from the results found. Similarly, [6] uses five different parameters- predictive accuracy, error rate, training time, classification index and comprehensibility- to compare 7 different classification algorithms. However, since the algorithms are tested on only four different datasets and the datasets are from different fields, the results are not as reliable.While only 7 binary classification datasets are used for comparison in [7], the research excels in its thorough exploration of hyper parameter tuning and performance metrics. In total, 2000 models with different hyperparameter tuning are tested on each dataset. The research shows how different performance metrics work best for different sorts of problems. Yet, hardly any focus has been given to the reasoning behind the methodology used to define the datasets that are being used.

Relevant works related to more niche problems are also analyzed. [8] compares algorithms for five imbalanced credit scoring datasets. To measure the extent of adverse effect on the predictive power of the algorithms, the class imbalance is increased gradually in the datasets.The results show that ensemble learners perform the best when it comes to dealing with imbalanced data. However, very few datasets are used and increasing the number of datasets of that specific field would have given more conclusive results for [8]. [9] compares 6 data mining algorithms. 57 classification sets are used, for which simple, statistical and information measures for each datasets are found. Then SOM to cluster the datasets. Five clusters are found and the performance of each cluster is analysed.

Finally, while most of the papers point out detailed observation about the results, they do not explain why the specific algorithm is behaving that way. In other words, there has been no attempt at connecting the patterns shown by the algorithms to the theory behind how the algorithm works. Without such explanations, understanding of the practical applications of the algorithms remain limited.

III. Background Study

A. Comparison of RF with GB and XGB

RF is an ensemble method that uses bootstrapping while GB and XGB use the concept of boosting to build a model. Due to majority voting of uncorrelated decision trees, RF is able to generalise the data much better, thus decreasing the variance [10]. As more uncorrelated trees are added, the variance further decreases for RF. However, when there is only a few important features in the dataset, most of the trees pick those features for splitting, and the trees end up correlated to each other. Due to random feature and sample subsampling, bias of RF increases because doing so imposes restrictions on the model and means it cannot take into account the complex relationships between features in the dataset. On the other hand, ensemble methods that use boosting such as GB and XGB, reduce overall error by reducing bias. This happens because the subsequent trees are trained on the errors of the previous trees and learn the mistakes that its predecessors have committed. As more and more trees are combined in the model additively, the difference between the correct target value of a sample and the expected prediction of the predictive model decreases, thus reducing the bias. However, as the subsequent trees are trained on the error of the previous model, the model might tend to overfit the dataset, thus increasing the variance. GB and especially XGB handles this problem of overfitting by adding regularization terms to the overall objective function, thus limiting the increase in variance of the model. Therefore, we can say that GB and XGB are likely to perform better than RF in general.

When there are a lot of noisy features in the dataset along with several important features, GB might work worse than RF [10]. This is because the trees built by GB will take all the features into account, including the noisy ones. This means a lot of trees will come up with inaccurate predictions and will mar the predictions of the overall model. However, this does not happen in RF and XGB due to its feature sub-sampling attribute which allows them to leave out the noisy features and only use the important ones for building the trees.

RF needs to build many uncorrelated trees while training in order to reduce the variance of the model and to achieve a competitive accuracy level while prediction. Thus, computation complexity of RF tends to be higher than GB and XGB.

B. Comparison of GB and XGB

We can understand from the algorithm for XGB that, if the Hessian is equals to 1, the tree structures learnt by XGB and GB are the same [10]. Moreover, GB will then be able to learn better leaf weights. Such an instance occurs for the square loss (where hessian becomes 1), and thus GB is likely to perform better than XGB for the square loss [10].

XGB includes feature subsampling along with row subsampling, while GB only allows row subsampling. Subsampling allows a better generalisation of the machine learning problem. As all the features are not taken into account while determining the best split, it cannot learn all the specific relations of the problem which might result in overfitting.

XGB can also handle missing values better. Whenever it faces a missing value it automatically learns the best direction to move according to a sparsity aware split finding algorithm. On the other hand, the sklearn implementation of gradient boosting cannot handle missing values and requires imputation before the data can be processed by the algorithm.

XGB also introduces a penalization term that is added to the loss function to make the ultimate objective function which we want to minimize. This further reduces the chance of overfitting [10].

IV. Clustering metrics

A. Simple metrics

1) Number of samples: The number of samples is the number of observations in the dataset. The higher the sample size, the more data the algorithm has to train on, and thus should be able to approximate the underlying distribution better.

2) Total number of Features: These are the number of attributes that the dataset has, using which the algorithm will be able to make a prediction.

3) Ratio-missingvalues-totalvalues: For our analysis, we calculated the ratio of the number of missing values to the total number of values in the dataset. The total number of values is found by multiplying rows into columns. This method of taking the ratio ensures that the number of missing values is not dependent on the sample size of the dataset.

4) Ratio-min-to-max-target: The min-to-max-target ratio shows the class distribution of the target variable. The majority values is considered the majority target and the minority values is called the minority target.

B. Statistical Measure

1) standard variation of Mahalanobis distance: The Mahalanobis distance (MD) [12] is the distance between two points in multivariate space. If two or more variables are correlated, the axes are no longer at right angles, and the measurements between the points using euclidean distance becomes inaccurate. The MD solves this problem by taking correlation between multiple variables into consideration. The equation of MD is given by

$$D^2 = (x - m)^T \cdot C^{-1} \cdot (x - m) \tag{1}$$

2) Generalized variance or Determinant: The determinant [13] of a matrix can be viewed as the volume scaling factor of the linear transformation described by the matrix i.e. how much the function is stretching or compressing regions of space near a point. Thus the determinant of the covariance matrix of a dataset gives the spreadness or multivariate variance of the whole dataset.

3) Maximum eigen value: The maximum eigenvalues of the co-variance matrix of a dataset encodes the variability of the dataset in an orthogonal basis that captures most of its spreadness (a.k.a the first principle component basis).

4) Skewness: Skewness is a measure of the asymmetry of the distribution of a real valued random variable about its mean. The coefficient of skewness represents the asymmetry of one dimensional distributions. It is calculated on the basis of the first three moments of the distribution. [14] Mardias Skewness [15] is the most popular way of measuring multivariate skewness. This is done by comparing the joint distribution of several variables against a multivariate normal distribution. Mardia defined multivariate skewness

$$b_{1,p} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[(\mathbf{x}_i - \overline{\mathbf{x}})' \, \mathbf{S}^{-1} \, (\mathbf{x}_j - \overline{\mathbf{x}}) \right]^3 \tag{2}$$

where x is a p X 1 vector of random variables and S is the biased sample covariance matrix of x defined as

$$\mathbf{S} = \frac{1}{n} \sum_{i=1}^{n} \left[(\mathbf{x}_i - \overline{\mathbf{x}}) \, (\mathbf{x}_i - \overline{\mathbf{x}})' \right] \tag{3}$$

5) Kurtosis: Kurtosis is a measure of the heaviness of the tail of the distribution of a real-valued random variable. [15] It is the normalised fourth central moment of a distribution minus 3. For multivariate data analysis, the joint distribution of several variables against a multivariate normal distribution are compared. Mardias kurtosis is given as follows:

$$b_{2,p} = \frac{1}{n} \sum_{i=1}^{n} \left[(\mathbf{x}_i - \overline{\mathbf{x}})' \, \mathbf{S}^{-1} \, (\mathbf{x}_i - \overline{\mathbf{x}}) \right]^2 \tag{4}$$

where x is a p X 1 vector of random variables and S is the biased sample co-variance matrix of x defined as

$$\mathbf{S} = \frac{1}{n} \sum_{i=1}^{n} \left[(\mathbf{x}_i - \overline{\mathbf{x}}) \, (\mathbf{x}_i - \overline{\mathbf{x}})' \right] \tag{5}$$

C. Information Theory

1) Entropy: Entropy helps us define the dataset at hand using a different type of metric other than the simple and statistical measures, giving us the average rate of information produced by all the features in the dataset. The entropy equation is as follows:

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{6}$$

where S is the series consisting the values of a certain feature, c is the total number of unique classes in the feature, and p_i is the probability value of the i^{th} class in the series S.
Taking the geometric mean over all the entropy values of all the features gives us a global entropy measure of the dataset. The equation can be written as follows:

$$\overline{E}_{global} = \sqrt[n]{\prod_{i=1}^{n} E(S)_i} = n\sqrt{E(S)_1 \cdot E(S)_2 \ldots\ldots E(S)_n} \tag{7}$$

where $\overline{E}_{global}$ is the global entropy measure for the dataset, n is the total features of the dataset, and $E(S)_i$ is the entropy

value of the i^{th} feature. The geometric mean was especially used to aggregate the entropy values because entropy values are ratios and geometric mean is more suited to calculate the average of ratios.

D. Feature-target Relationship

1) PCA: PCA reduces the dimension of a dataset by squashing it onto a proper lower-dimensional line (or more generally a hyperplane) which retains as much of the original datas defining characteristics as possible.
We apply PCA on the dataset and extract the top two principle components (kept constant for all datasets). Then, we fit a logistic regression algorithm on the top components and extract the coefficients of the regression equation. This is done in order to capture the relationship between the features and the target of a dataset.

2) Logistic regression coefficients: Linear regression model works [16] by drawing a best fit line that follows the data trend as closely as possible. The model finds out the best fit line by calculating the mean, standard deviation and correlation coefficient and forming the equation in the form of $y = m_1x_1 + m_2x_2 + ... + m_nx_n + b$, which is a best fit hyperplane, In this case, n =2.
For binary classification datasets, we predict using logistic regression. This logistic regression equation if formed by substituting the value of the best fit line of $y = m_1x_1 + m_2x_2 + b$, into the sigmoid function $\frac{1}{1+e^{-y}}$. Thus the logistic regression equation becomes

$$p = \frac{1}{1 + e^{-m_1x_1+m_2x_2+b}} \tag{8}$$

where p is the probability. B is the y intercept.
The term m1 signifies that increasing x_1 by 1 increases the log-odds by 1.
The term m2 signifies that increasing x_2 by 1 increases the log-odds by 2.

V. Research Methodology

A. Pre-processing for calculating clustering metrics

Dropping the ID column of each dataset that had one, simple measures of the dataset are found. Then entropy is calculated. For entropy calculation, the target variable is dropped and missing values are removed by imputing median values or 0 depending on the feature description. Categorical features are unchanged for entropy calculation. Before multivariate statistical analysis is done on the datasets, the categorical features are one-hot encoded. Multivariate statistical analysis is carried out on the dataset without the target columns. To take into account the relationship between the features and the target variables, Principal Component Analysis (PCA) is used to reduce the dimensions of the dataset and a logistic regression algorithm is fitted on the PCA to extract the 3 coefficients of the regression equation, which we use as separate metrics to define the respective datasets. All the features are then formatted into a dataframe, with each row representing one dataset.

A heatmap is generated from the dataset that we created, to find out the correlation between the features. This is done as a part of feature selection technique to remove one of any highly correlated pairs of features as they will affect the clustering algorithm in the same way and thus will be a redundant feature. Feature pairs with correlation coefficient higher than 0.8 are considered to be highly correlated. Then, among the two features, only one is kept. The ratio of missing values is also dropped because the values are far too low (value is 0 in most cases) to make a significant difference in the clustering. Finally, the metrics selected for clustering are number-of-samples, total features, maxEigen, mahala-std, coefficient-1, coefficient-2, intercept, and entropy.

B. Clustering Specifications

We choose the SOMWard method for the purpose of our experiment because it works better for multivariate inputs than Single Linkage [12]. Additionally, with the Ward method there is a possibility that the clusters might come out separated from each other, which the SOMWard accounts for, resulting in connected clusters [12]. While creating the clusters, priority of all the attributes are kept the same. A map with 7 nodes and automatic ratio (3:3) is trained. Training schedule Accurate with tension 0.5 is used.

After generating the map, we observed that the algorithm had produced 4 clusters. Moreover, we used the dendrogram as a second algorithm which produced 3 clusters. Thus, we restricted the number of clusters in Viscovery to 3, and eventually observed a better formation of the clusters- one that was closer to our hypothesis.

C. Performance metrics Specifications

Training Error: At first, the dataset (both the features and the target) is split into training and testing set in a 3:1 ratio in a stratified fashion. The algorithms are trained on the training set and evaluated on the testing set. The default parameters are used in each of the algorithms and a fixed random state is given so that we get the same results each time. Finally, we used 10-fold cross-validation with scoring parameter to be the negative log loss, and used the mean of the cross validation scores to compare between the algorithm's errors.

Training Time: The time is measured from when the algorithm starts training to when it finishes (known as the training time). The algorithm takes in the dataset as the parameter. The rows of the dataset starts from 10 and varied by 10 rows while keeping all other parameters the same. The training time is measured for each iteration and is saved in a list. A line graph is plotted to show how the training time varies with the rows of the dataset for each of the algorithms. The computer specification used for the experiment was Intel core i7 CPU with 2.6 Ghz Processor, 8 GB RAM and a 64 bit Windows Operating System.

VI. Experimental Results and Pattern Analysis

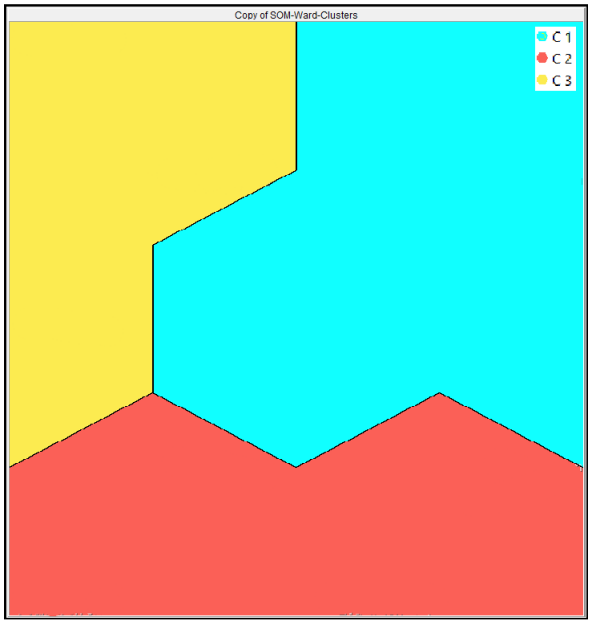

Fig. 1. Clustering result of Viscovery SOMine

The figure above illustrates the clustering results of Viscovery SOMine. The following tables, collected from Viscoverys data records, show datasets that were grouped together in the respective clusters by the SOMWard Algorithm.

A. Cross-validation Result

The cross-validation results are taken to 6 significant figures. If the first two significant figures of the results match, then we can say that the algorithms have similar performance in the dataset.

Dataset	Mean negative log loss			
	XGB	GB	RF10	RF 100
Heart	0.446154	0.498642	0.988785	0.411308
Liver	0.578399	0.586842	0.908296	0.533793
Medical cost	0.259432	0.283841	1.279313	0.309199
Pima	0.494139	0.515483	1.015469	0.481629
Surgical	0.215011	0.217052	0.512348	0.294460
Wisconsin	0.109580	0.140430	0.279839	0.127403
Default credit card	0.453466	0.453767	1.238930	0.485627
German credit	0.564577	0.565521	0.786191	0.570210
Acute Liver	0.190661	0.194885	0.574527	0.249274
Titanic	0.562271	0.564511	1.866634	0.815905
Wilt	0.019205	0.024629	0.059887	0.021034
Income	0.296315	0.296179	0.868255	0.361468
Santander	0.274733	0.279031	0.930741	0.288843

TABLE I
Cross-validation values of each algorithm on each dataset

Datasets		First	Second	Third	Fourth
C1	Liver	RF100	XGB	GB	RF10
	German Credit	XGB	GB	RF100	RF10
	Medical Cost	XGB	GB	RF100	RF10
	Wilt	XGB	RF100	GB	RF10
	Pima	RF100	XGB	GB	RF10
C2	Surgical	XGB	≃GB	RF100	RF10
	Wisconsin	XGB	RF100	GB	RF10
	Titanic	XGB	≃ GB	RF100	RF10
	Heart	RF100	XGB	GB	RF10
C3	Acute Liver	XGB	≃ GB	RF100	RF10
	Default Credit Card	XGB	≃ GB	RF100	RF10
	Santander	XGB	≃ GB	RF100	RF10
	Income	GB	≃ XGB	RF100	RF10

TABLE II
Clustering and ranking algorithms according to Viscovery

1) Cluster 1:

- For 10 trees, XGB is the best, followed by GB and RF respectively
- For 100 trees, RF is the best performing for Liver and Pima. For Medical cost and German credit, XGB works the best, followed by GB. However, for wilt, RF is better than GB

With small number of trees in RF, it can be concluded that RF is the worst performing one. However, as the number of trees increases, the performance of RF for this cluster gets better. For two of the datasets, it overtakes XGB, and for one it only overtakes GB. Perhaps, with increasing trees, RF might outperform XGB for this cluster. This is because we know from theory that as we add more uncorrelated trees to the forests, the variance decreases and consequently the error decreases. In other words, Random forests tend to generalise better as the number of trees increases.

2) Cluster 2:

- For 10 trees, XGB is the best, followed by GB and RF for Heart and Wisconsin. Moreover, for Surgical and Titanic, XGB=GB(their negative log loss is equal to two decimal places) followed by RF
- For 100 trees, Rf is the best performing algorithm for Heart. For wisconsin, XGB is the best followed by RF and GB. And, for Surgical and Titanic, XGB=GB.

With a small number of trees for RF, XGB is the best performing in all datasets. Performance of RF increases with increasing number of trees for this cluster as well- for Heart, it outperforms XGB and for Wisconsin, it outperforms GB. It could be predicted, that with increasing number of trees, RF would outperform XGB for Wisconsin dataset. However, for Surgical and Titanic, the performance of XGB and GB are much better than RF, even if the trees are increased.

This cluster in general show weak patterns in terms of accuracy and so, further research is needed to come up with generalised conclusive results.

3) Cluster 3:

- for 10 trees XGB=GB for all of them, with RF being the least accurate

- for 100 trees as well, XGB=GB for all of them, with RF being the least accurate

Therefore we can conclude, for those in cluster 3, RF is the worst choice in terms of accuracy. With increasing number of trees, the accuracy does get better but both XGB and GB outperforms RF in both the cases. Using XGB or GB gives similar performance results.

4) General observations: XGB works the best in general. This is because in terms of the algorithms, XGB calculates the hessian at each iteration, and uses both the gradient and hessian to calculate the Gain for determining the best split. Using the hessian, XGB is able to better approximate the loss function. This means that XGB can come up with a better tree structure as it calculates the Gain more accurately. On the other hand, the GB algorithm only calculates the gradient of the loss function and uses it to calculate Gain at each iteration which means its Gain measure is less accurate than that of XGB.

B. Time Complexity Result

We plot a line graph of training time against sample size (number of rows) to see how training time changes according to the sample size for each dataset. These are then grouped according to the clusters found using SOM.

1) Cluster 1:

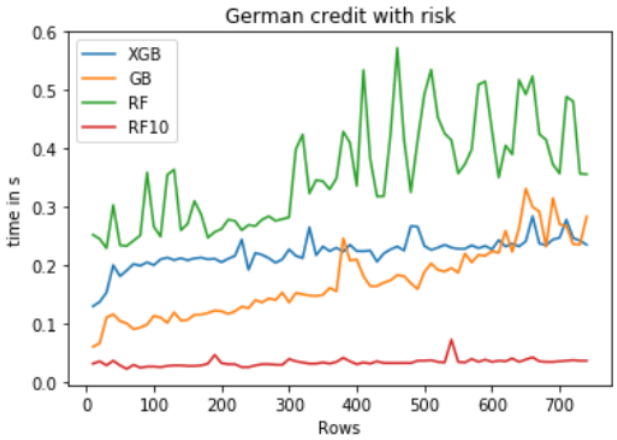

Fig. 2. Time graph for German Credit dataset

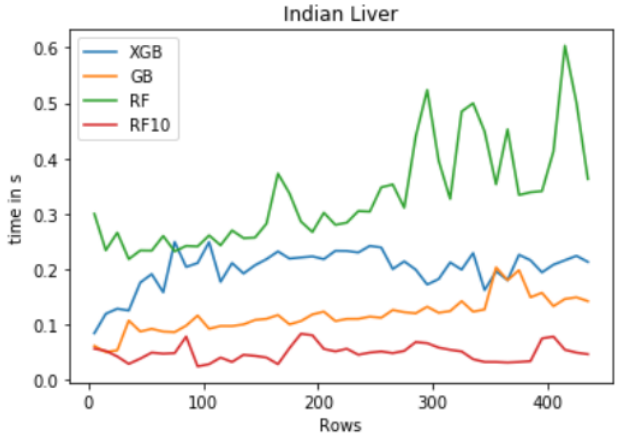

Fig. 3. Time graph for Indian Liver dataset

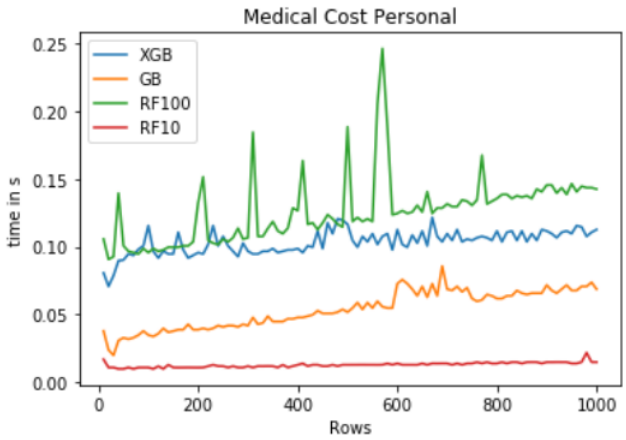

Fig. 4. Time graph for Medical cost dataset

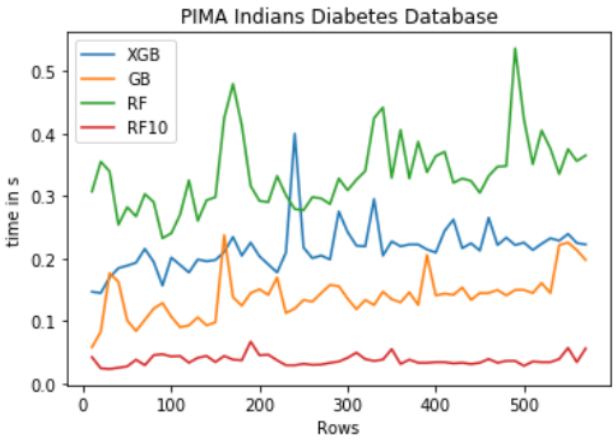

Fig. 5. Time graph for Pima dataset

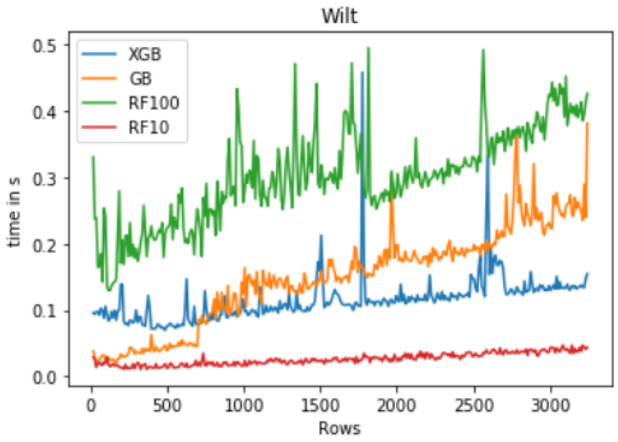

Fig. 6. Time graph for Wilt dataset

- For cluster 1, for most of the datasets GB seems to always have a lower training time than XGB. However , for German Credit dataset, GB seems to fluctuate around XGB for sample numbers greater than 600.
- For Wilt, GB overtakes XGB for samples over 700.
- In the case of RF, for all datasets RF with 100 trees takes the longest time to train, while RF with 10 trees takes the least.

2) Cluster 2:

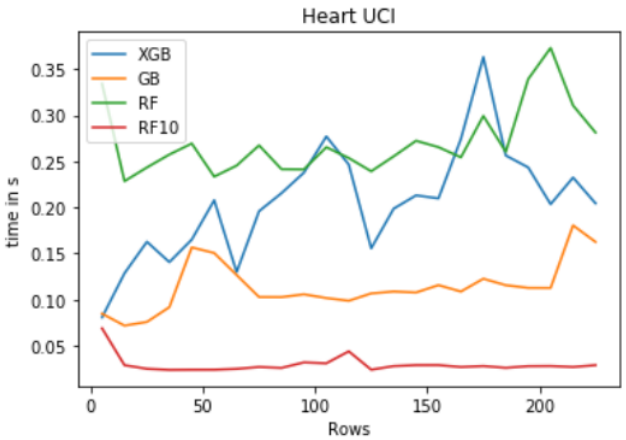

Fig. 7. Time graph for Heart dataset

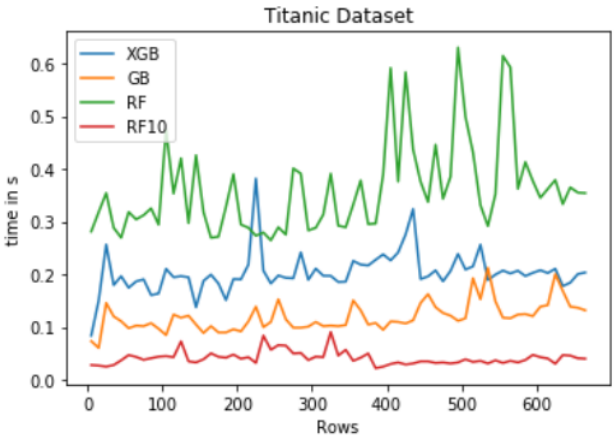

Fig. 8. Time graph for Titanic dataset

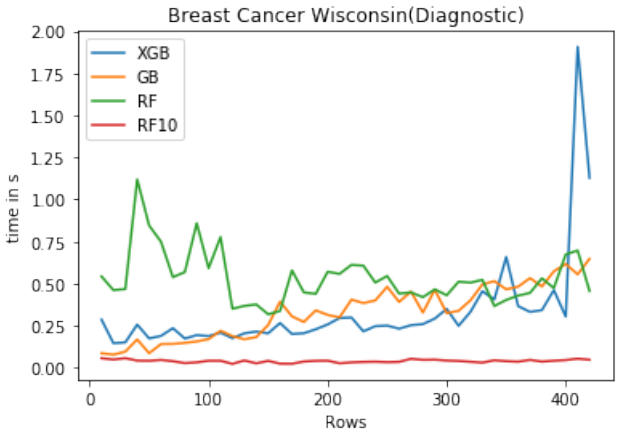

Fig. 9. Time graph for Breast Cancer Wisconsin dataset

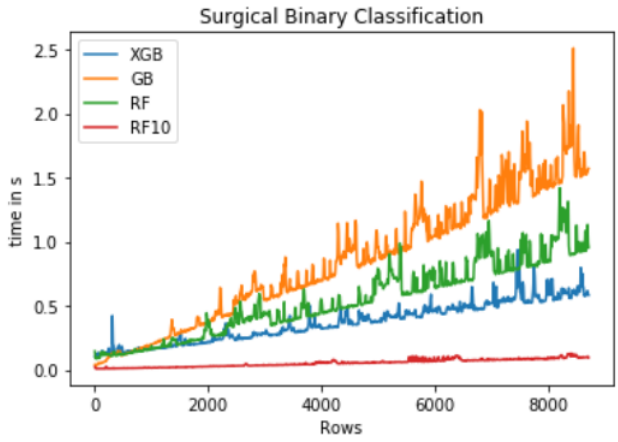

Fig. 10. Time graph for Surgical Classification dataset

- RF with 10 trees always takes less time than the other algorithms for all datasets in this cluster.
- For Heart and Titanic, GB curve always stays below XGB, and for Breast Cancer Wisconsin and Surgical,

GB overtakes XGB after around 120 samples and 430 samples respectively.

- For smaller samples, RF100 takes longer than the other algorithms. GB takes the longest time for surgical. Surgical also has the highest number of samples. It can be inferred that even for the other datasets, if sample was higher, GB would overtake RF100.
- For surgical, which has a large number of samples, we observe that RF-100 initially takes less time than XGB but as the number of samples increase, RF100 eventually takes over XGB.
- The rate at which GB increases is much higher than the rates at which other algorithms increase in time complexity, specifically for when the number of samples is higher such as in surgical.

3) Cluster 3:

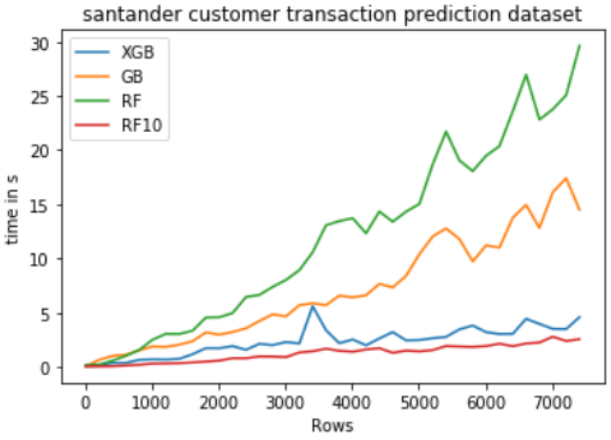

Fig. 11. Time graph for Santander customer transaction prediction dataset

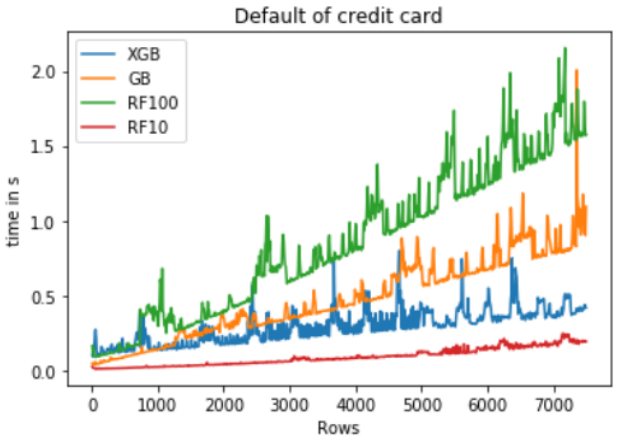

Fig. 12. Time graph for default credit card dataset

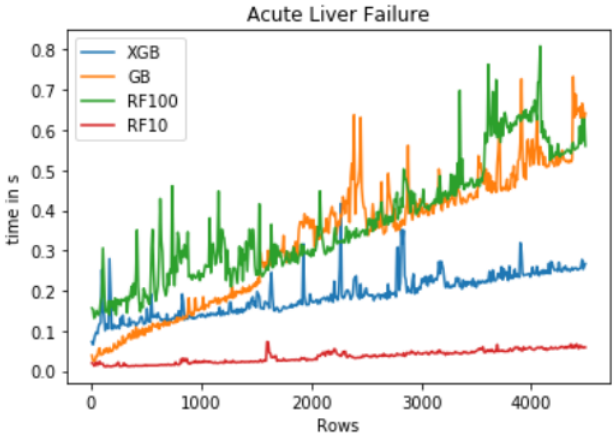

Fig. 13. Time graph for acute liver failure dataset

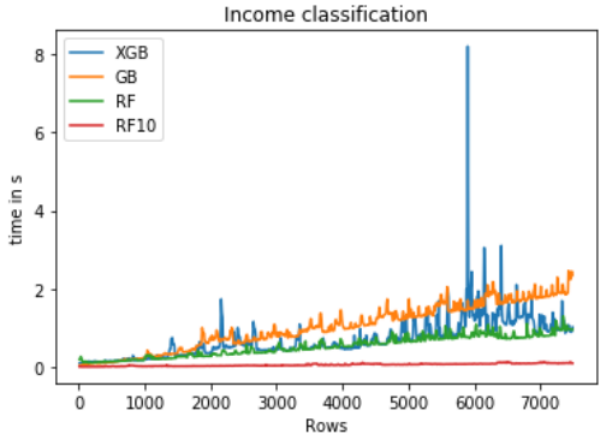

Fig. 14. Time graph for income classification dataset

- RF with 10 trees always takes less time than the other algorithms for all datasets in this cluster.
- GB overtakes XGB and increases at a much faster rate than XGB, as the number of samples increase. For income and acute liver, GB overtakes at around 800 samples, and for default credit it overtakes at around 1000 samples. The only exception for this is santander dataset,where GB overtakes at 50 samples.
- GB overtakes RF-100 for acute liver and income classification, but for default credit and santander, RF100 takes longer. So, there it cannot be conclusively said which algorithm will take the longest in this cluster.
- As the sample size gets larger, XGB increases very little. It increases linearly at a lower gradient than GB. One possible reason for this could be that GB uses a recursive function to calculate leaf weights and XGB uses a equation that includes the gradient and the Hessian.

4) General Observations:

- From theory, it is known that while calculating the leaf weights, GB uses a recursive function that carries out an exhaustive search through all possible values of leaf

weights and returns the set of weights that minimizes the loss function. On the other hand, XGB uses a equation that includes the gradient and the Hessian. Therefore, with increasing sample size, time complexity of GB increases at a faster rate and overtakes XGB, which increases at a much slower rate.

- In general, XGB is the most efficient in terms of time complexity as the number of samples increase.
- At higher number of samples, the training time for RF-100 increases polynomially.

VII. Rule Generation

Using the patterns found in the cluster, a ranking order for the algorithms is generated. The algorithms are ranked in order from highest to lowest in terms of accuracy and also in terms of time complexity. The one with the highest accuracy is scored 4 and the lowest accuracy is scored 1. The one that takes the highest training time at the original number of samples is scored 1, and the one that takes the least time is given a score of 4. Here, 4 is the best score an algorithm can get at any performance metrics. However, if two algorithms approximately (equal if rounded to two decimal places) have equal performance in terms of negative log loss, then an average of both their scores are taken to be their final scores.i.e they both get equal scores.

After the algorithms are ranked for each of the datasets, the cumulative points for all the datasets in a specific cluster are calculated and then divided by the number of datasets in that cluster. This is done for both accuracy and time complexity. The tables below illustrate the calculations explained above.

Cluster 1								
Accuracy					Time			
Dataset	XGB	GB	RF100	RF10	XGB	GB	RF100	RF10
Liver	3	2	4	1	2	3	1	4
German Credit	4	3	2	1	3	2	1	4
Medical Cost	4	3	2	1	2	3	1	4
Wilt	4	2	3	1	3	2	1	4
Pima	3	2	4	1	2	3	1	4
Cumulative	18	12	15	5	12	13	5	20
Average	3.6	2.4	3	1	2.4	2.6	1	4

TABLE III
Algorithm score for cluster 1

Cluster 2								
Accuracy					Time			
Dataset	XGB	GB	RF100	RF10	XGB	GB	RF100	RF10
Surgical	3.5	3.5	2	1	3	1	2	4
Wisconsin	4	2	3	1	1	2	3	4
Titanic	3.5	3.5	2	1	2	3	1	4
Heart	3	2	4	1	2	3	1	4
Cumulative	14	11	11	4	8	9	7	16
Average	3.5	2.75	2.75	1	2	2.25	1.75	4

TABLE IV
Algorithm score for cluster 2

Cluster 3								
Accuracy					Time			
Dataset	XGB	GB	RF100	RF10	XGB	GB	RF100	RF10
Acute liver	3.5	3.5	2	1	3	2	1	4
Default Credit	3.5	3.5	2	1	3	2	1	4
Santander	3.5	3.5	2	1	3	2	1	4
Income	3.5	3.5	2	1	3	1	2	4
Cumulative	14	14	8	4	12	7	5	6
Average	3.5	3.5	2	1	3	1.75	1.25	4

TABLE V
Algorithm score for cluster 3

Finally, we get two specific scores for an algorithm for a specific cluster- one for accuracy and another for time complexity. For a specific cluster, these scores are considered constants. Using these scores, we can separately rank the algorithms in terms of just one performance metric. This is shown below.

Algorithms	Cluster1		Cluster2		Cluster3	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
XGB	1st	3rd	1st	3rd	1st	2nd
GB	3rd	2nd	2nd	2nd	1st	3rd
RF100	2nd	4th	2nd	4th	3rd	4th
RF10	4th	1st	4th	1st	4th	1st

TABLE VI
Algorithm Ranking

Moreover, we can also rank the algorithms in terms of both the performance metrics. Giving them equal weights, an overall rank for each of the cluster can be found by adding the two scores and dividing it by two. Mathematically, this can be written as $y = 0.5 * (accuracyscore) + 0.5 * (timecomplexityscore)$. By doing this for each of the clusters, the ranking of the algorithms is shown below.

Algorithms	Cluster1 combined	Cluster2 combined	Cluster3 combined
XGB	1st	1st	1st
GB	2nd	2nd	2nd
RF100	4th	4th	3rd
RF10	2nd	2nd	4th

TABLE VII
Combined Algorithm Ranking

The above formula can be generalised to give an equation, where the weights of the performance metrics are to be tuned. Mathematically speaking, this can be written as $y = ax_1 + bx_2$, where x_1 is the accuracy score and x_2 is the time complexity score, which are constant for a specific cluster. The weight of accuracy is a, and weight of time complexity is b, where $a + b = 1$. The user can determine what weight they want to give to accuracy and time complexity according to their neccessities, i.e. they can tune a and b. For example, if a user requires an algorithm to be fast but can do with an approximate prediction, then b is increased and a is decreased. Then using the formula, the score for each of the algorithms is calculated. These can then be ranked according to what has the highest score. The higher the score, the better the algorithm should be for that specific problem.

A. Example using different weight parameters

Suppose there are two users who come with datasets that after clustering is found to be part of cluster one. However,

for one, the weight of performance criteria are $a = 0.3$ and $b = 0.7$, i.e. the user requires an algorithm that gives a decent approximate prediction but trains on data really fast. Then, by using the scores we have found upon clustering, the total score is calculated as:

$$XGB = 0.3 * 3.6 + 0.7 * 2.4 = 2.76$$

$$GB = 0.3 * 2.4 + 0.7 * 2.6 = 2.54$$

$$RF100 = 0.3 * 3 + 0.7 * 1 = 1.6$$

$$RF10 = 0.3 * 1 + 0.7 * 4 = 3.1$$

Therefore for this cluster, RF10 would be the best.

The ranks according to this weight is given below. This example also shows that even the worst performing algorithms can sometimes be the best suited algorithm for a specific problem, as we have learnt from the No-free lunch theorem.

Algorithms	Cluster1 combined
XGB	2nd
GB	3rd
RF100	4th
RF10	1st

TABLE VIII
COMBINED ALGORITHM RANKING WITH HIGHER WEIGHT FOR TIME

On the other hand, the criteria of the user with the other dataset is $a = 0.7$ and $b = 0.3$, i.e the user requires an algorithm that is highly accurate and can take a long time to train. Using the scores we have found, upon clustering the dataset into cluster one, then the total score would be:

$$XGB = 0.7 * 3.6 + 0.3 * 2.4 = 3.24$$

$$GB = 0.7 * 2.4 + 0.3 * 2.6 = 2.46$$

$$RF100 = 0.7 * 3 + 0.3 * 1 = 2.4$$

$$RF10 = 0.7 * 1 + 0.3 * 4 = 1.9$$

Therefore for this cluster, XGB would be the best.The ranks according to this weight is given below:

Algorithms	Cluster1 combined
XGB	1st
GB	2nd
RF100	3rd
RF10	4th

TABLE IX
COMBINED ALGORITHM RANKING WITH HIGHER WEIGHT FOR ACCURACY

B. Generalised equation for scoring algorithms

In fact, if the performance metrics were increased, this rule can be further expanded to accommodate new parameters. Generalizing the term mathematically gives

$$y_j = a_1x_1 + a_2x_2 + a_3x_3 + ... + a_nx_n \tag{9}$$

.

such that $a_1 + a_2 + ... + a_n = 1$ and where, y_j is the total score for the j^{th} algorithm, a_i is the weight for each performance metrics, x_j is the constant score for the j^{th} algorithm, and for the i^{th} performance metrics for a specific cluster.

VIII. CONCLUSION AND FUTURE WORK

Our research analyses the implementation of Random Forest, Gradient Boosting, and Extreme Gradient Boosting in binary classification problems. The datasets are described using simple measures, multivariate statistical measures and information theory measures and then, cluster analysis is done to group the datasets. The accuracy and time complexity of the ensemble methods are then calculated, compared and analysed. According to the results, algorithms work similarly for datasets in the same cluster with the exception of some anomalies. Finally, using the results of the algorithms from the clusters, a scoring method is proposed to rank the algorithms in terms of accuracy and complexity. This method is then defined mathematically, thus giving us a generalised rule to find what is the best suited algorithm for a specific problem according to the users requirement, given the weights of the performance metrics are provided. Hence, our proposed method works by mapping a complex problem into a simpler one, and then solving that to find out which algorithm will be the best for a specific problem.

Further works related to this can include tuning hyperparameters to find the optimum conditions for the specific datasets. For random forest, the optimal number of trees could be found, and for gradient boosting and for XGB, hyperparameters such as learning rate can be tuned. Additionally, the number of datasets can be increased to have more robust clusters and more accurate scores for each of the algorithms specific to the clusters that we generated. A similar methodology used in this research can be extended to regression problems, multiclass classification problems and so on. Additionally, a more comprehensive study could be done by incorporating other types of algorithms such as Support Vector Machine, Naive Bayes, Neural Networks and so on. The performance metrics used can also be increased to include other accuracy measures, and subjective measures may also be taken into account for a more detailed study.

REFERENCES

[1] T. O. Ayodele, Types of machine learning algorithms, inNew Advancesin Machine Learning, Y. Zhang, Ed., Rijeka: IntechOpen, 2010, ch. 3.doi:10.5772/9385. [Online]. Available: https://doi.org/10.5772/9385.

[2] Shalev-Shwartz, S., Ben-David, S. (2014). Understanding machine learning:

[3] R. D. KING, C. FENG, and A. SUTHERLAND, Statlog: Comparison ofclassification algorithms on large real-world problems,Applied Artificial In-telligence, vol. 9, no. 3, pp. 289333, 1995.doi: 10.1080/08839519508945477.eprint: https : / / doi . org / 10 . 1080 / 08839519508945477. [Online]. Available:https://doi.org/10.1080/08839519508945477.

[4] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, A comparison of prediction accuracy,complexity, and training time of thirty-three old and new classification al-gorithms,Machine Learning, vol. 40, no. 3, pp. 203228, Sep. 2000,issn:1573-0565.doi: 10.1023/A:1007608224229. [Online]. Available: https://doi.org/10.1023/A:1007608224229.

[5] S. Ali and K. A. Smith, On learning algorithm selection for classification,Appl. Soft Comput., vol. 6, no. 2, pp. 119138, Jan. 2006,issn: 1568-4946.doi: 10.1016/j.asoc.2004.12.002. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2004.12.002.

[6] L. Zhou and K. K. Lai, Benchmarking binary classification models on datasets with different degrees of imbalance,Frontiers of Computer Science inChina, vol. 3, no. 2, pp. 205216, Jun. 2009,issn: 1673-7466.doi: 10.1007/s11704-009-0027-1. [Online]. Available: https://doi.org/10.1007/s11704-009-0027-1.

[7] R. Caruana and A. Niculescu-Mizil, An empirical comparison of supervisedlearning algorithms, inProceedings of the 23rd International Conference onMachine Learning, ser. ICML 06, Pittsburgh, Pennsylvania, USA: ACM, 2006,pp. 161168,isbn: 1-59593-383-2.doi: 10 . 1145 / 1143844 . 1143865. [Online].Available: http://doi.acm.org/10.1145/1143844.1143865.

[8] I. Brown and C. Mues, An experimental comparison of classification algo-rithms for imbalanced credit scoring data sets,Expert Syst. Appl., vol. 39,no. 3, pp. 34463453, Feb. 2012,issn: 0957-4174.doi: 10.1016/j.eswa.2011.09.033. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2011.09.033.

[9] 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). Physica, Heidelberg.

[10] Hastie, T., Tibshirani, R., Friedman, J., Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer, 27(2), 83-85.

[11] V. SOMine,Viscovery somine users manual, English, Viscovery SOMine,275 pp., published

[12] A. Lahav, R. Talmon, and Y. Kluger, Mahalanobis distance informed byclustering,Information and Inference: A Journal of the IMA, vol. 8, no. 2,pp. 377406, Aug. 2018,issn: 2049-8772.doi: 10.1093/imaiai/iay011. eprint:http://oup.prod.sis.lan/imaiai/article-pdf/8/2/377/28864971/iay011.pdf.[Online]. Available: https://doi.org/10.1093/imaiai/iay011.

[13] M. Hubert and M. Debruyne, Minimum covariance determinant,Wiley in-terdisciplinary reviews: Computational statistics, vol. 2, no. 1, pp. 3643, 2010.

[14] D.Dori c, E. Nikoli c-Dori c, V. Jevremovi c, and J. Malisi c, On measuringskewness and kurtosis,Quality and Quantity, vol. 43, no. 3, pp. 481493,2009.

[15] K. V. Mardia, Measures of multivariate skewness and kurtosis with applica-tions,Biometrika, vol. 57, no. 3, pp. 519530, 1970

[16] Menard, S. (2002). Applied logistic regression analysis (Vol. 106). Sage.

[17] D. Dua and C. Graff,UCI machine learning repository, 2017. [Online]. Avail-able: http://archive.ics.uci.edu/ml.

[18] I.-C. Yeh and C.-h. Lien, The comparisons of data mining techniques forthe predictive accuracy of probability of default of credit card clients,ExpertSystems with Applications, vol. 36, no. 2, pp. 24732480, 2009

[19] B. A. Johnson, R. Tateishi, and N. T. Hoan, A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pineand oak trees,International journal of remote sensing, vol. 34, no. 20, pp. 69696982, 2013.

[20] D. Dua and C. Graff,UCI machine learning repository, 2017. [Online]. Avail-able: http://archive.ics.uci.edu/ml.

[21] R. Kohavi, Scaling up the accuracy of naive-bayes classifiers: A decision-treehybrid., inKdd, Citeseer, vol. 96, 1996, pp. 202207

[22] M. Lichman,UCI machine learning repository, 2013. [Online]. Available: http://archive.ics.uci.edu/ml.

[23] J. P. Eaton, Haas charles a,TitanicTriumph and Tragedy, 1994.

[24] Available at https://www.kaggle.com/c/santander-customer-satisfaction(2019/08/06).

[25] P. Hind, Encyclopedia titanica,Online. Internet. Available http://atschool.eduweb. co. uk/phind, 1999.

[26] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, Usingthe adap learning algorithm to forecast the onset of diabetes mellitus, inProceedings of the Annual Symposium on Computer Application in MedicalCare, American Medical Informatics Association, 1988, p. 261

[27] B. Lantz,Machine learning with R. Packt Publishing Ltd, 2013.