Kirkuk University Journal /Scientific Studies (KUJSS)
Volume 15, Issue 2, June 2020, pp. (25-44)
ISSN: 1992-0849 (Print), 2616-6801 (Online)

Study of Data Mining Algorithms Using a Dataset from the Size-Effect on Open Source Software Defects

Muthana Yaseen Nawaf[1], Maaeda Mohsin Rashid[2]
[1, 2] Department of Computer Science, College of Computer Science and Information Technology, Kirkuk University, Kirkuk, Iraq.
[1]muthana2085@yahoo.com, [2]maeda_hiderm@yahoo.com

Abstract

This article focuses on the quality of data mining algorithms in terms of the accuracy ratio and time consumption. So, in order to figure out the best algorithm among the classification and clustering algorithms, the WEKA program will be testing all algorithms using a real dataset from the size effect on defect proneness for open source software. The Mozilla product is adopted as an example of open source software. The dataset that is used in this paper represents the output of the study of the size effect on defect proneness in the open source software. The study of Mozilla product shows a significant relationship between the size of software and the number of defect proneness in software. The Mozilla product study produced a dataset to be as inputs of the WEKA program in order to compare the data mining tools (algorithms). We use the Naive Bayes, Decision Trees J48, Expectation-maximization for classifying and K-Star and Simple KMeans for clustering methods. The findings demonstrate the difference between the algorithms according to the accuracy, and the time consuming to reach the result in each algorithm. Furthermore, the effect of the software size is significant on defect proneness. Finally, the experiments are conducted in WEKA with the aim of this research is finding out the best algorithm in terms of accuracy and time-consuming. At the end, the paper will be figuring out the best algorithm in order to choose and depending on it in the tests of classification and clustering.

Keywords: Data mining algorithms, Weka, defect proneness, Mozilla products, open source software.

DOI: http://doi.org/10.32894/kujss.2020.15.2.3

Kirkuk University Journal /Scientific Studies (KUJSS)
Volume 15, Issue 2, June 2020, pp. (25-44)
ISSN: 1992-0849 (Print), 2616-6801 (Online)

دراسة خوارزميات تنقيب البيانات باستخدام مجموعة بيانات من تأثير الحجم على عيوب البرمجيات مفتوحة المصدر

مثنى ياسين نواف[1]، مائدة محسن رشيد[2]

[1,2] قسم علوم الحاسوب، كلية علوم الحاسبات وتكنولوجيا المعلومات، جامعة كركوك، كركوك، العراق.

[1]Muthana2085@yahoo.com, [2]maeda_hiderm@yahoo.com

الملخص

تركز هذه المقالة على جودة خوارزميات التنقيب عن البيانات من حيث نسبة الدقة واستهلاك الوقت. لذلك، لمعرفة أفضل خوارزمية بين خوارزميات التصنيف والتجميع، سيقوم برنامج WEKA باختبار جميع الخوارزميات باستخدام مجموعة بيانات حقيقية من تأثير الحجم على قابلية الخلل في برمجيات المصدر المفتوح. إن دراسة منتج موزيلا تظهر علاقة مهمة بين حجم البرنامج وعدد عيوب الخلل في البرمجيات. أنتجت دراسة منتج Mozilla مجموعة بيانات تمثل مدخلات برنامج WEKA لمقارنة أدوات استخراج البيانات (الخوارزميات) باستخدام Naive Bayes و Decision Trees J48 و Expatiation Maximization (EM) للتصنيف و K-Star و Simple KMeans لطرق التجميع. بعد كل شيء، تظهر نتيجة هذا البحث الفرق بين الخوارزميات وفقًا للدقة، والوقت المستهلك للوصول إلى النتيجة الأخيرة في كل خوارزمية. أيضا، تأثير حجم البرنامج مهم على ظهور العيوب. أجريت جميع التجارب في WEKA بهدف البحث عن أفضل خوارزمية من حيث الدقة والوقت المستغرق. ورقة البحث تحدد في النهاية الخوارزمية الافضل لاعتمادها في اختبارات التصنيف وايجاد اوجه التشابه والاختلاف في البيانات المعتمدة.

الكلمات الدالة: خوارزميات استخراج البيانات، Weka، العيوب، منتجات Mozilla، البرامج مفتوحة المصدر.

DOI: http://doi.org/10.32894/kujss.2020.15.2.3

1. Introduction:

According to Fitzgerald, (2006), the open source software (OSS) which is a computerized software with its main source code readily available for exploration, distort and distribute the same for whatever purposes. The software must be supported by a valid license in which the holder of the copyright. Its development usually takes place in a rather public collaborative manner and is the most vivid example defining open-source development and is more often listed in comparison to open content evolution as legally defined or content as defined technically.

It has been reported by the Standish group (2008) that the consumer has managed to save close to 60 billion dollars annually through accommodation of the open-source software models. The use of Weka open-source software as a basis for experiments on classification has efficiently increased the chances of reproduction by other researchers. The public domain can now access the module created on the current work to date. A minor cataloguing error rate measuring less than 5.4% was able to be achieved using instance-based learner.

A. The corpus: The corpus a composition of 15545 illustrations, hosting 6 traits each portraying a binary character to indicate how a defect in the same can be fixed. The meanings of the characters are illustrated as pointed out below:

1. Id – this is a distinct numeric identification allocated to different C++ category.
2. Start – this is a time infinitesimally more superior than the duration of modification in which conclusion was derived.
3. End – either the duration of the following modification, or the finalization of the conclusion period, or erasing.
4. Event – it is set to 1 upon fixing of a defect at the period indicated by the end.
5. Erasing of a class is much easier when progressing to a conclusion whose event is based on 1 if the category is erased for corrective maintenance.
6. Size – it is dependent duration covariate, and its pillars host the number of source lines codes of the C++ at the duration of starting.
7. State – it transforms to 1 after initially being set at 0. This occurs after the category undergoes an event, and thereafter retains as 1.

8. The tool used for the categorizing our experiments was the Weka toolkit in which at the start of the test the tactic induced was five-fold cross-validation with a defaulted parameter set to have a wide-ranging synopsis.

B. Classification Algorithms

1. Lazy k star: K is a case-based categorizer that is the category of a test relying upon the type of those training cases that are familiar to it as indicated by some comparison utilities. It is different from other learners by the fact of using entropy-based distance utilities.

2. Naïve Bayes Classifier: The classifier group belongs to common classifiers relying upon probabilities based on using the theory of Bayes with critical assumptions regarding the features (Wang and Li, 2015). Since its extensive study in the 1950s, the theory remains a popular tactic for text classification with the difficulty of analyzing documents aligning to one class or the other, with features being indicated by word frequencies (Puga and Altman, 2015). Its competitiveness is based upon its advanced tactics for support vector equipment. It also retrieves applications in automated medical diagnosis. The naïve Bayes model is a provisional possibility character. When issued with a situation instance to be categorized represented by dependable variables, it allocates to the instance possibilities for each of K expected results or categories. The concern of the formulation is when there is a huge number of features is huge or can accommodate a large number of values; then it becomes challenging to base such a model on possibilities tables of feasibilities, thus reformulating the model to shape it into a more tractable classifier. Remotely the naive Bayes method is one of the conditional probability models whereby in a problem instance classification is done using a vector $x = (x_1, \dots \dots x_n)$. There are n features represented by the vector which are all dependent variables (Cox et al., 2016) ,This method assigns vectors to probabilities $P(C_k \backslash X_1, \dots \dots ., X_n)$ for every value of K. The scenario is done using the following formula:

$$P(C_k \backslash \mathrm{x}) = \frac{P(C_k)\, P(x \backslash C_k)}{\mathrm{P}_x} \tag{1}$$

In layman language and at the same time applying the Bayesian probability terminologies, the equation can be re-written as follows:

$$Posterior = \frac{\text{Prior} \times \text{Likelihood}}{\text{Evidence}} \tag{2}$$

Practically, the interest should only be in the numerator of the fraction where in this case the denominator fails to depend on C, and the values of the features F_i are given. Consequently, achieving a constant number of denominators. Therefore, the numerator equals to the following joint probability :

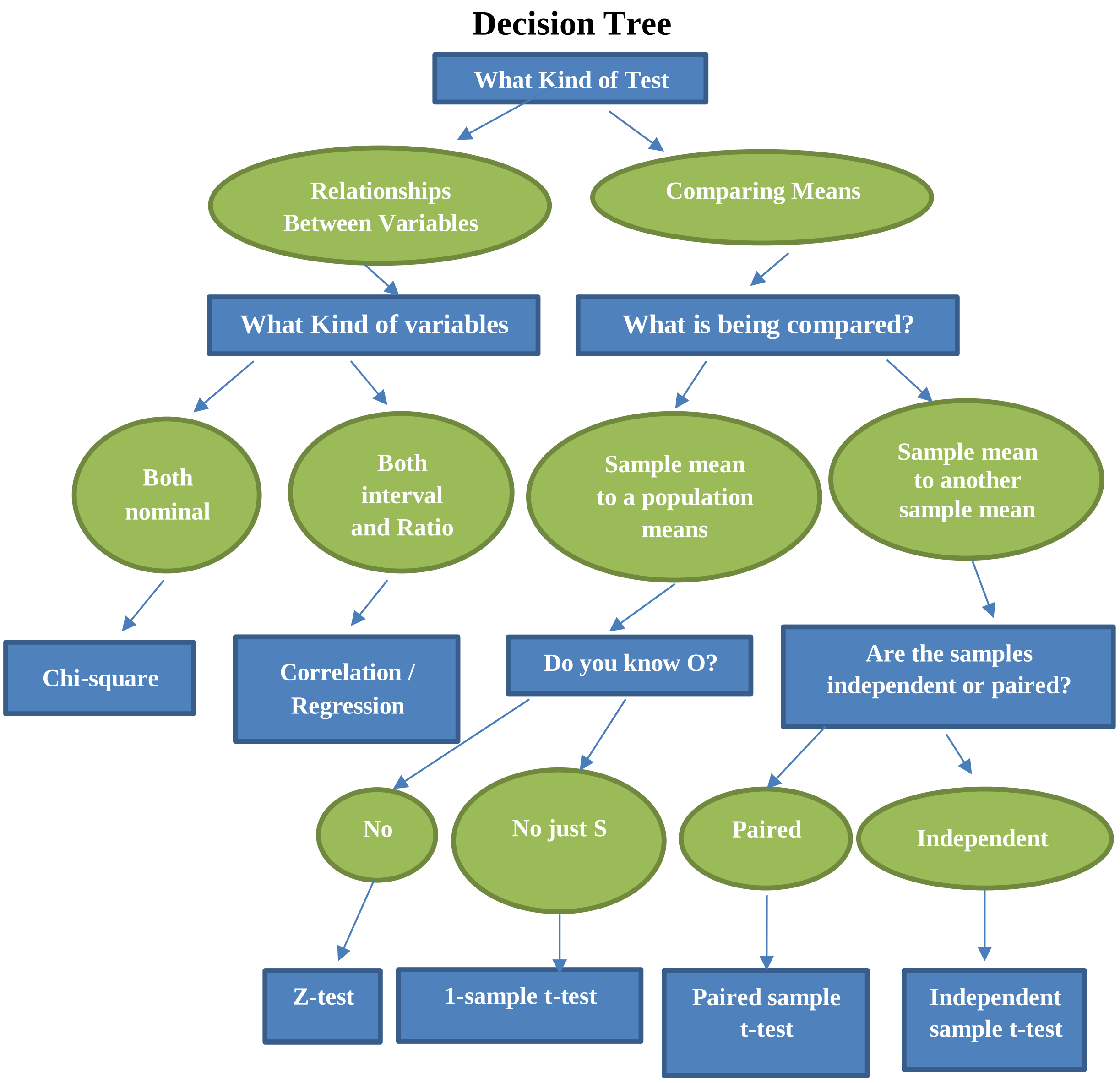
$$P(C_k, X_1, \dots \dots, X_n) \tag{3}$$

3. Decision Tree: Dai and Ji (2014) argued that during probability analysis, the decision tree is viewed as a critical decision tool support another tool that makes use of a tree-like graph or specifically a model of decisions as well as any likely consequence that include some chance event outcomes (Wu et al., 2015). Additionally, the rest consequences included utility and resource costs. Therefore, the decision tree is one of the ways of displaying an algorithm. In most cases, decision trees are implemented in solving operations research problems. In other words, the use of decision analysis when identifying a strategy will most likely get to a goal.

In terms of appearance, a decision tree looks like a flowchart structure whereby every internal node point to a test that is an attribute such as to indicate whether a head /tail comes up when a coin is tossed. Every branch of the tree indicates the result of the experiment or occurrence. On the other hand, leaf nodes showed a class label that is the decision taken after considering every possible scenario. The classification rules are represented by the path pointing from root to leaf. When carrying out decision analysis, the decision tree, as well as the similar influence diagram, are both applied like visual or analytical decision support tools. In such a case, the expected values (which also refer to the anticipated utility of all the competing alternatives) are computed. In normal conditions, the decision tree contents three kinds of nodes:

1. Decision: It is usually pointed out using squares.
2. Chance: It is mostly represented using circles.
3. End: It is depicted using triangles.

In most cases, decision trees are applied during research. Furthermore, decision trees are useful when carrying out analysis in order to locate a strategy that is most likely to result in a specific goal. Practically, when decisions must be taken online without any consideration to incomplete knowledge, the decision tree is supposed to be accompanied by a probability model. The method can be applied as the best alternative and an online selection model algorithm. Furthermore, to calculate conditional probabilities, decision trees are also implemented as a descriptive means. Decision trees are believed to have a strong influence on diagrams and utility functions. Additionally, they can be applied via other decision analysis tools methods as taught in undergraduate level in business, public health schools and health economics. In such cases, operations research and management science methods are applied. From the outcome, the path as drawn manually using traditional methods is as shown below in Fig. 1.



4. Clustering: (Dhana Chandra et al., 2015) suggested that the cluster analysis (which in other words, is just referred to as clustering) relates to grouping set of objects to depict objects found in the same group. The grouping is known as a cluster and is found to be similar in one way another. The clusters are more related to each other as compared to the ones found in other groups.

Clustering is mostly used in data exploratory and mining, which is a common technique applied in statistical data analysis. It is also used in other fields, like machine learning, computer vision, info retrieval, and finally bioinformatics. Specifically, cluster analysis is one of the specific algorithms, whereby general tasks are solved. The solution is achieved through algorithms differing significantly in terms of what comprises a cluster as well as how to efficiently locate clusters. The most common cluster notions consist of groups bearing minute distances amongst cluster groups and found in dense areas in the data space (Zheng et al., 2015). The intervals are found in certain statistical distributions. Therefore, clustering is formulated as a multi-objective optimization approach.

The most effective clustering algorithm that also works as a parameter setting, including values like distance function, intensity threshold and the number of the clusters is dependent on an individual dataset and the expected application of the outcomes. However, cluster analysis is not an automatic operation; instead, it is an iterative procedure with a lot of knowledge discovery. The process also has an interactive multi objective optimization involving both trial and failure. Finally, in most cases, it is critical to modify both model parameters and data preprocessing until the intended results are achieved bearing of the desired properties.

C- Expectation-Maximization (EM): According to the study of Nilashi et al., (2015), applications of EM algorithm include calculating the maximum probability parameters in a statistical model. Such an application applies in cases where it is difficult to solve equations directly. In the solving, the equations the models use latent variables and unknown parameters plus known data observations. The implication is that either the data has missing values, or it is possible to reformat the model in a simpler way by making assumptions that there exist additional data points that are not observed. To illustrate this an example of a mixture is given a simpler description by presuming that every observation (Data Point) bears a corresponding unobserved data point. There also exists a latent variable that specifies a mixture component belonging to every data point.

To calculate the maximum probability solution requires the use of derivatives of the probability function while considering every unknown value viz. both the parameters as well as the latent variables. Additionally, it involves solving the resultant equations simultaneously. However, this is not usually possible in the case of statistical models having latent variables. Instead, it results in a set of overlapping equations whereby the parameter solution must have the exact values of the latent variables and vice-versa. However, during substitution one set of equations, when it is substituted in the other, it leads to an equation that cannot be solved. The scenario in Fig.2 and Fig.3 provide examples on EM clustering.

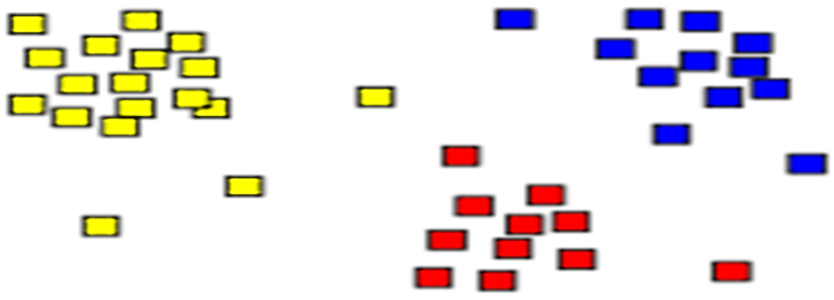


Fig.2: Showing Gaussian-distributed data, and how EM works.

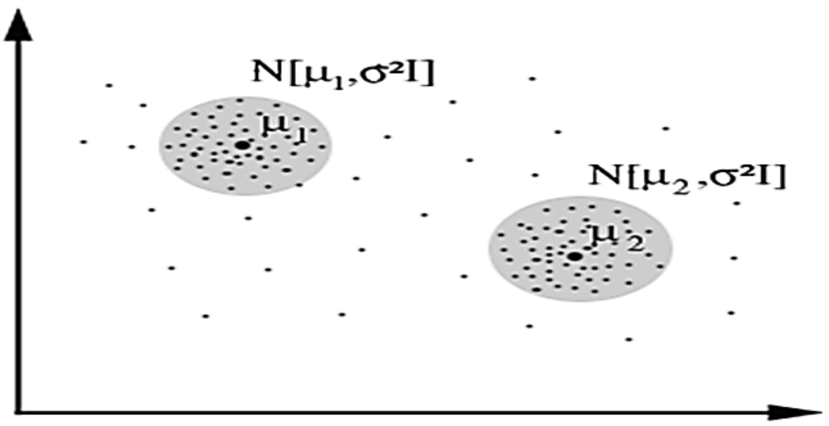


Fig.3: Showing Gaussian-distribution can’t be used to modeling density-based cluster.

5. **K mean:** K-means clustering is one of the vector quantization methods. The method originated from signal processing, which is a popular criterion for analyzing data during mining. The K-means clustering method objective is to partition and observe k clusters where each observation is assigned to a cluster bearing the nearest mean and serves as the cluster model. The exercise culminates to data partitioning into the Voronoi cells data space.

One of the problems involved in this is that it is computationally difficult (NP-hard). Nevertheless, most efficient times, heuristic algorithms are mostly used to assemble quickly in a local optimum. The process resembles the expectation-maximization algorithm, which is a combination of Gaussian distribution through iterative refinement method used by the two algorithms. also, both processes apply cluster centers to effectively model data. However, the k-means clustering method finds comparable spatial clusters. However, the expected-maximization mechanism permits clusters to bear alternative shapes.

2. Result and Experiments:

A. Use of Classification Algorithms:

1. **Lazy K-star**: After applying the K star algorithm, the result is as in Fig.4, Fig.5 and Table1.

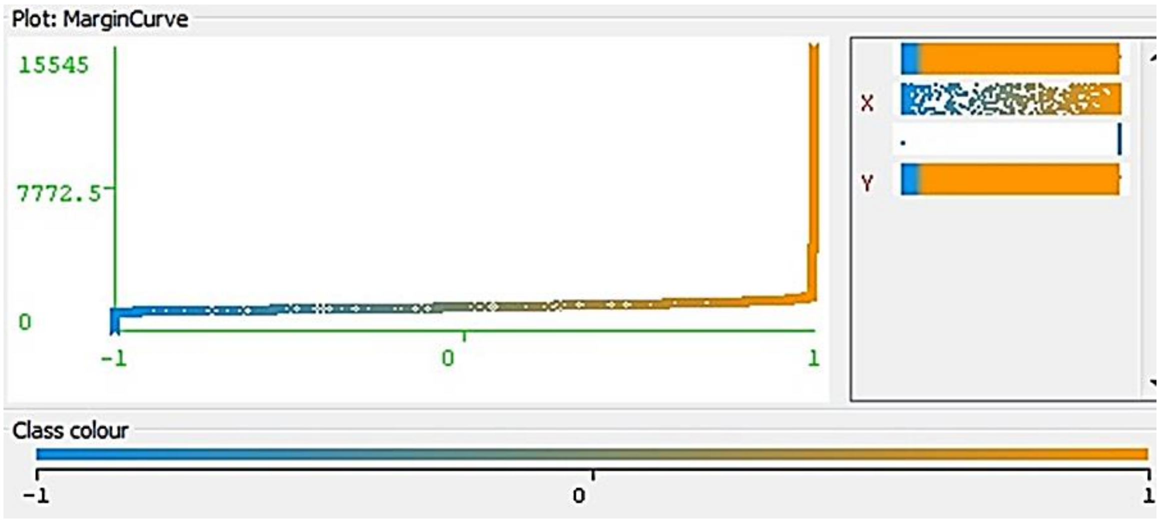


Fig.4: K-star visualization and Margin Curve.

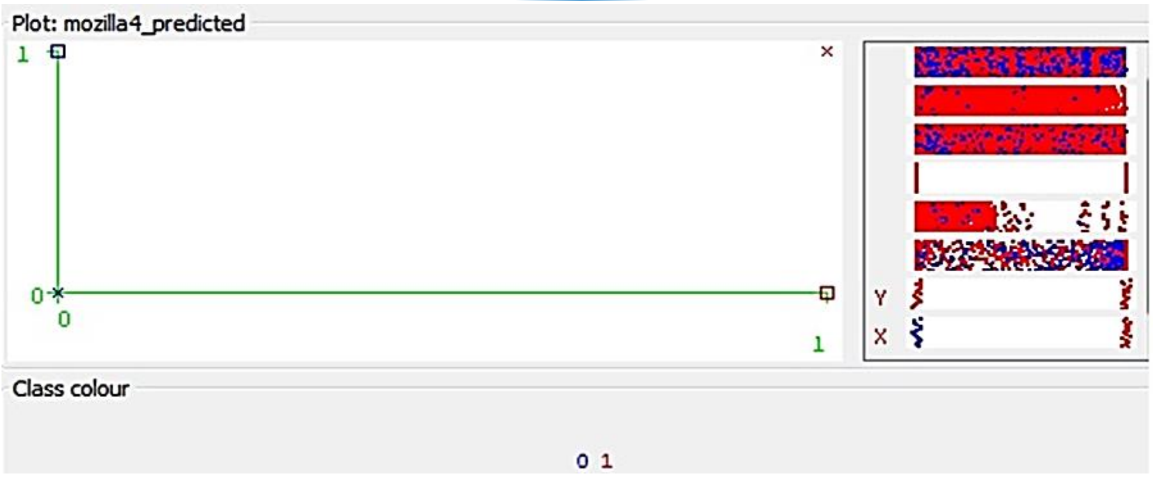


Fig.5: K-star visualizing and Classifier Error .

Table 1: Result in terms of k-star algorithm.

Scheme	WEKA. Classifiers .lazy .K-Star -B 20 -M a	
Relation	mozilla4	
Instances	15545	
Attributes	6	
Test mode	5-fold cross-validation	
Building model time	0.01 seconds	
Correct Classified	14297	91.9 %
Incorrect Classified	1248	8.0 %

2. **Naive Bayes Classifier:** The following results show Naive Bayes Classifier result as indicated in Fig.6, Fig.7 and Table 2.

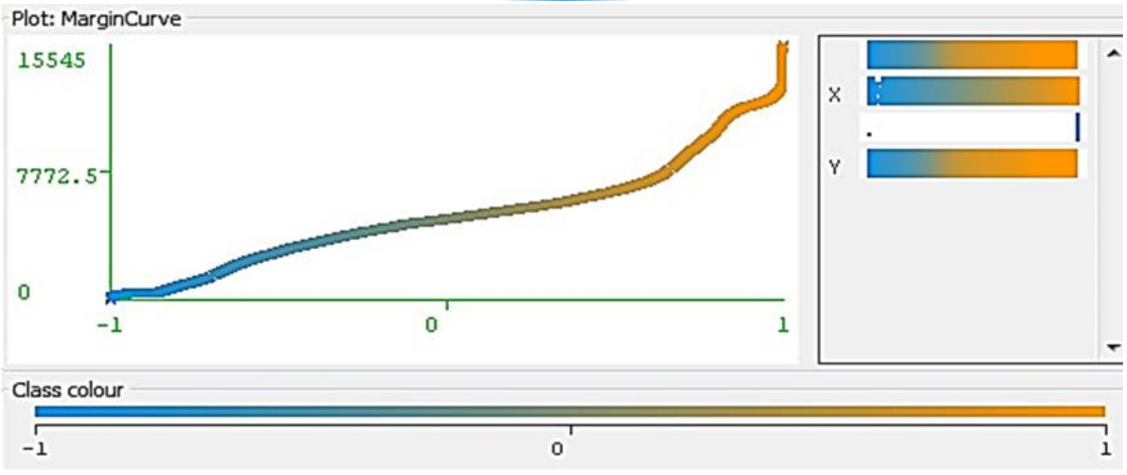


Fig.6: Showing Naïve Bayes visualization and Margin Curve.

Plot: mozilla4_predicted
1
0
0
1
Y
X
Class colour
0 1

Fig.7: Showing Naïve Bayes visualize and Classifier Error.

Table 2: The result in Naïve Bayes.

Scheme	WEKA.Classifiers. NaiveBayes	
Relation	mozilla4	
Instances	15545	
Attributes	6	
Test mode	5-fold cross-validation	
Building model time	0.06 seconds	
Correct Classified	10671	68.6 %
Incorrect Classified	4874	31.3 %

3. **Decision Tree:** When Decision Tree J48 Classifier is applied, the result will appear as in Fig.8, Fig.9, Fig.10 and Table 3.

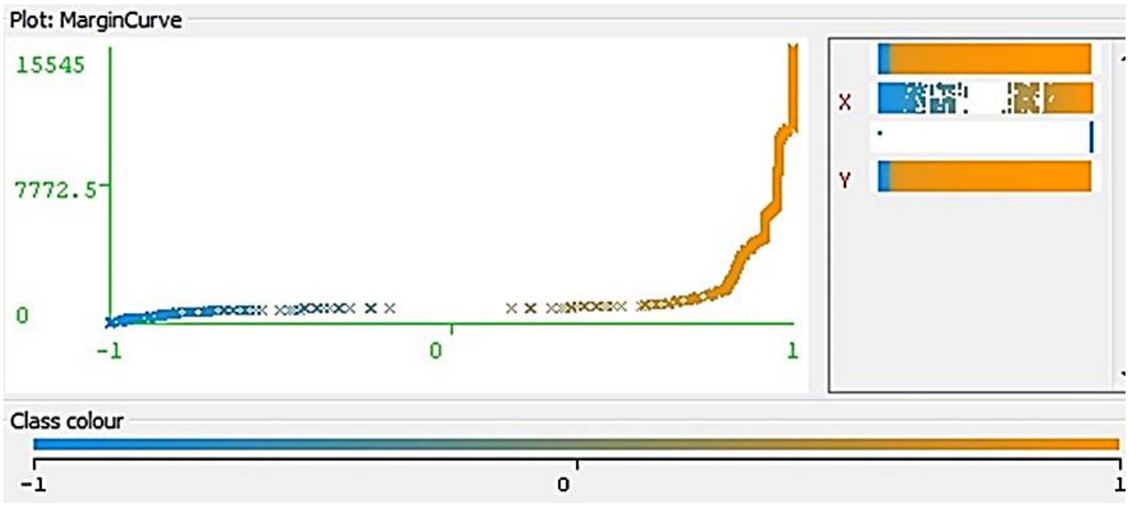


Fig.8: Decision Tree visualize: Margin Curve.

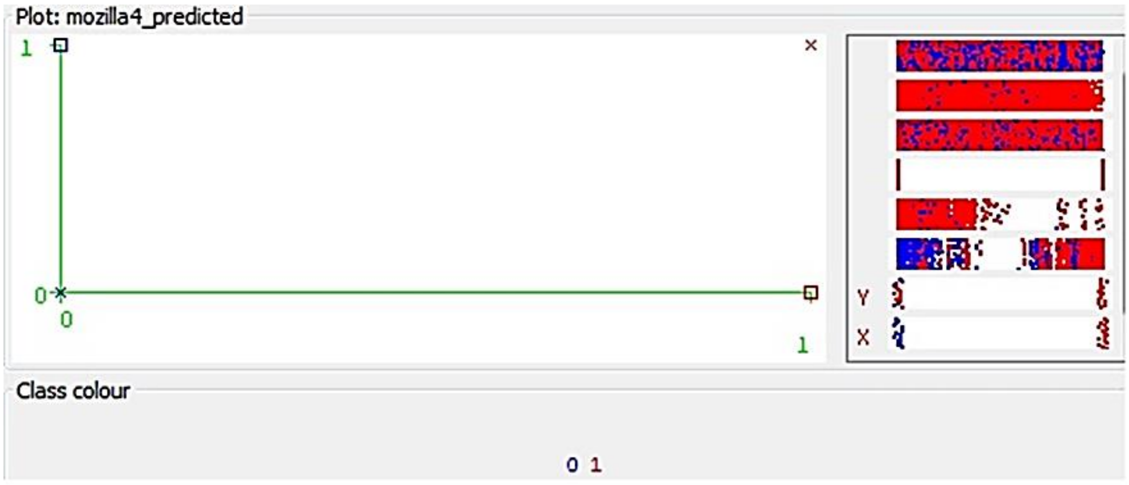


Fig.9: Decision Tree visualize: Classifier Error.

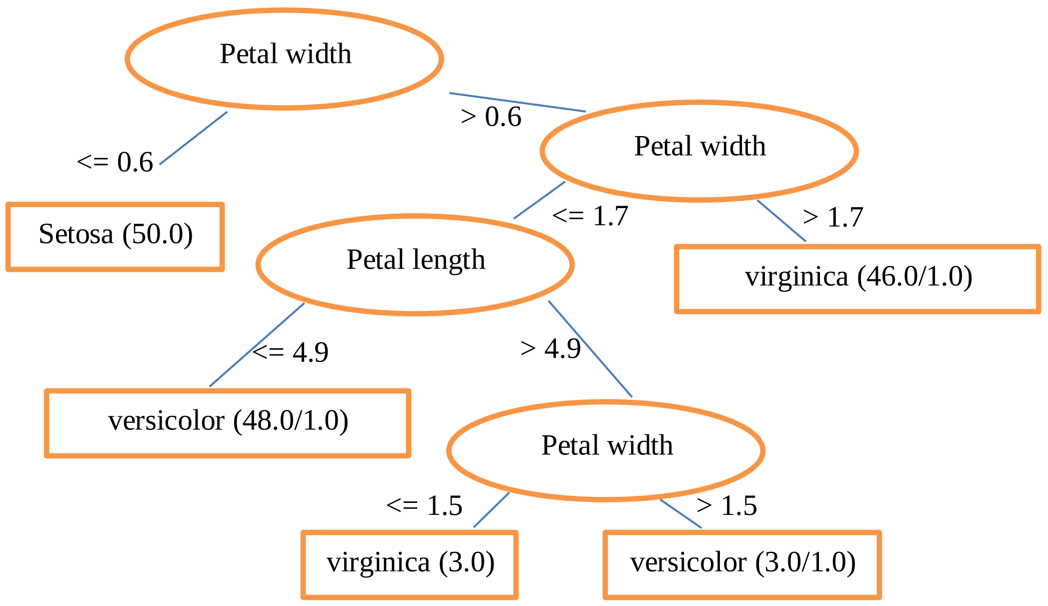


Fig. 10: Decision Tree J48 view tree (number of leaves 122 sizes of tree 243).

Table 3: The result in the Decision Tree.

Scheme	WEKA. classifiers. Trees. J48 -C 0.26 -M 2	
Relation	mozilla4	
Instances	10045	
Attributes	5	
Test mode	5-fold cross-validation	
Building model time	0.46 seconds	
Correct Classified	1500	94.7 %
Incorrect Classified	824	4.4 %

B. Applying Clustering Algorithms:

1. **Expectation-Maximization (EM)**: After implementing the EM cluster algorithm, the result is as in Fig.11 and Table 4.

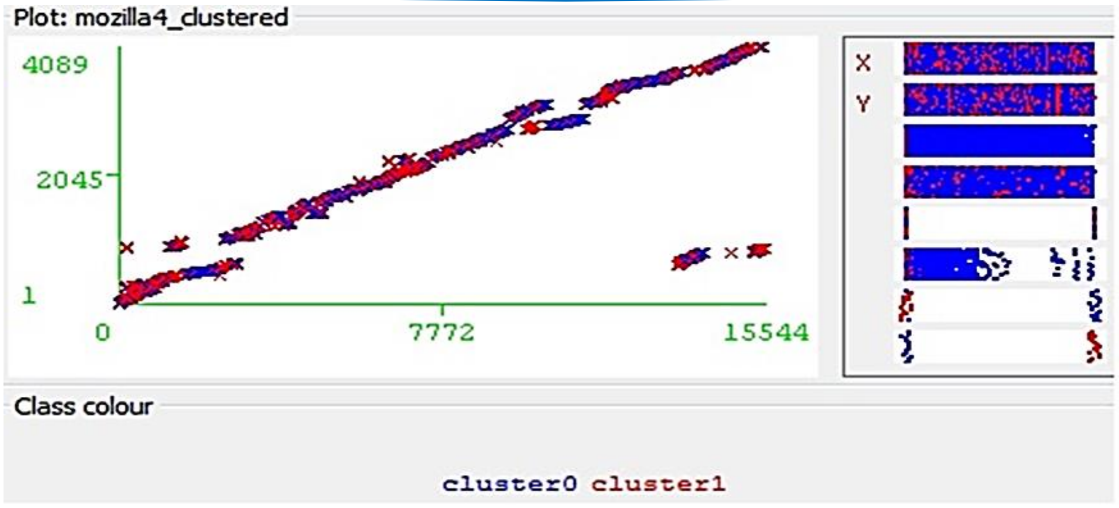


Fig.11: EM visualizing the cluster assignments.

Table 4: The result in the EM .

Scheme	weka. clusters. EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.1E-6 -num-slots 1 -S 101	
Relation	Mozilla5	
Instances	1045	
Attributes	6	
Test mode	evaluate on training data	
Building model time	100.01 seconds	
Clustered Instances		10516 (73.7%)
Clustered Instances		4028 (27%)

2. **K-Mean Cluster Algorithm:** After applying the K-mean cluster algorithm the following results are realized. see Fig.12 and Table 4.

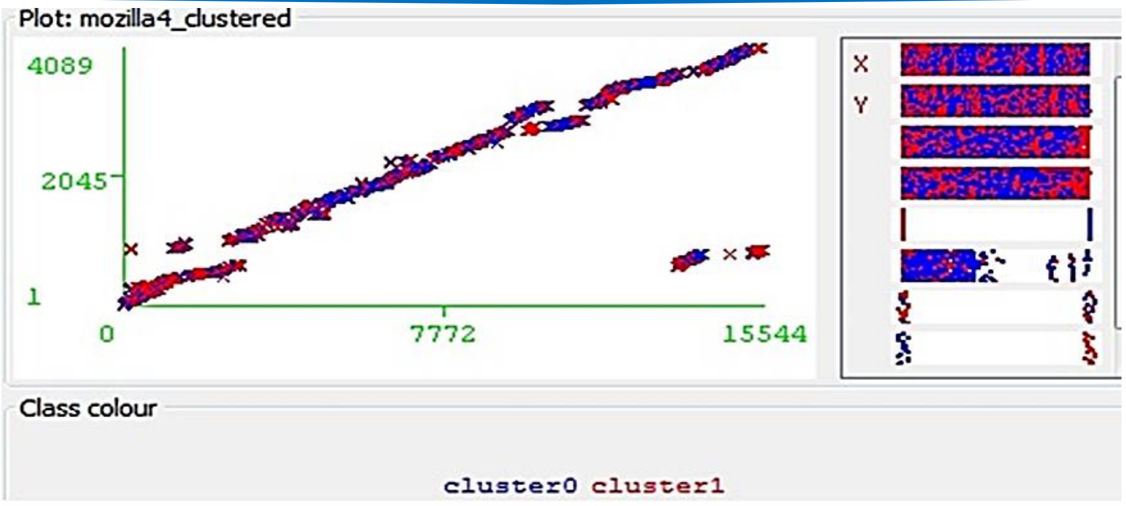


Fig.12: Simple K-mean visualize cluster assignments.

Table 5: Showing result in Simple K-Means.

Scheme	WEKA.clusterers.SimpleKMeans -init 0 -max-candidates 80 -periodic-pruning 90000 -min-density 2.4 -t1 -1.45 -t2 -1.2 -N 2 -Aweka.core.EuclideanDistance -R first-last -I 400 -num-slots 1 -S 12	
Relation	Mozilla3	
Instances	16345	
Attributes	5	
Test mode	evaluate on training data	
Building model time	0.18 seconds	
Clustered Instances	0	8929 (56%)
Clustered Instances	1	6416 (45%)

3. Results and Discussion:

In the experiments that were performed based on Mozilla open source software dataset, the lazy k-star result, which was represented in Fig. 4, Fig. 5 and Table 1. Where Fig. 4 clarify the correct and incorrect class in the tested data (15545 Instances) which was 92% correct classified and 8.0% incorrect, and test mode is 5-fold cross-validation as Table 1 showed. The Fig.5 showed the distribution of the attributes (6 attributes) between negative and positive tested as 0 (Negative) and 1 (Positive) for the classifier error. The building model time through lazy k-star algorithm was 0.01 seconds only where represents good performed from where the time consumption. The Naive Bayes Classifier result was represented in Fig. 6, Fig. 7 and Table 2. Where Fig. 6 showing correct and incorrect class in the tested data (15545 Instances) which was 68.6 % correct classified and 31.3% incorrect, and test mode is 5-fold cross-validation as Table 2 showed. The Fig. 7 showed the distribution of the attributes (6 attributes) between negative and positive tested as 0 (Negative) and 1 (Positive) for the classifier error.

The performance of the second algorithm compared with the first one was decline, because of the low accuracy of the correct class (68.6%) and the consumption of the time was more (0.06 second). The third algorithm result (Decision Tree) showed in Fig. 8, Fig. 9, and Table 3. Where Fig. 8 clarify the correct and incorrect class in the tested data (15545 Instances) which was 94.6 % correct classified and 5.3% incorrect, and test mode is 5-fold cross-validation as Table 3 showed. The Fig. 9 showed the distribution of the attributes (6 attributes) between negative and positive tested as 0 (Negative) and 1 (Positive) for the classifier error. Accordingly, to the accuracy, the decision tree algorithm performance was good, but with more time consumption (0.44 seconds).

The second part of the experiments which were represents applying the clustering algorithms in order to show the similarities and differences between the data used, EM algorithm result represented in Fig. 11 and Table 4. Where Fig. 11 visualizing the cluster assignments for Mozilla tested data (15545 Instances) which was cluster 0 (74% similar) and cluster 1 (26% differ), and test mode is evaluate on training data as Table 4 showed. The time consumption was too long compared to previous algorithms (102 seconds). K-mean algorithm result represented in Fig. 12 and Table 5. Where Fig. 12 visualizing the cluster assignments for Mozilla tested data (15545 Instances) which was cluster 0 (57% similar) and cluster 1 (43% differ), and test mode is evaluate on training data as Table5 showed. Accordingly, to the time consumption, the performance of this algorithm was pretty good compared with EM algorithm (0.19 seconds).

Finally, the following tables (Table 6 and 7) demonstrates the performance of the algorithms compared to each other according to the time consumption and the accuracy.

Table 6: Classification Algorithms.

Classification Algorithms			
NO.	**Algorithm Name**	**Time consumption**	**Accuracy**
1.	Lazy K-star	0.01	92%
2.	Naïve Bayes Classifier	0.06	68.6%
3.	Decision Tree	0.44	94.6%

Table 7: Clustering Algorithms.

Clustering Algorithms				
NO.	**Algorithm Name**	**Time consumption**	**Similarity**	**Differences**
1.	EM	102.4 seconds	74%	26%
2.	K mean	0.19 seconds	57%	43%

4. Conclusion:

In this article, tests were carried out using the data applied in all the five algorithms, as stated in the WEKA application. The findings demonstrated, as Table 6 show, that the classified algorithm is Decision Tree J48 with a correct accuracy rate of 94.6% and 0.44 seconds needed time to build the model. Accordingly, the model represented the best algorithm to be found between the rest of the classification algorithms. Although of the accuracy rate which is done by the Decision Tree J48 was good, the Lazy K-star algorithm performed well in both cases of time consumption and accuracy rate Table 6. So, the Lazy K-star in the case of huge data will be the best option to choose among the rest of classification algorithms. Alternatively, the cluster algorithms represented by EM as well as the Simple K-Means resulted in the given results that listed in the Table 7. The performance of the EM algorithm compared to the K-Mean algorithm was better in the case of time consumption. So, the EM algorithm will be the best option to choose instead of K-Mean algorithm. Among the used methods (which include the Bayes Theorem) argues the following probability:

$P(A|B) = P(B|A)\ P(A)P(B)P(A|B)=P(B|A)P(A)P(B).$

The method also assumes that the class conditional P(B|A) P(B|A) is independent so you can have $P(B|A) =\prod P(BI|A)$.

References

[1] R. Cox, C. W. Revie, D. Hurnik, & J. Sanchez, **"*Use of Bayesian Belief Network techniques to explore the interaction of biosecurity practices on the probability of porcine disease occurrence in Canada*"**, Preventive Veterinary Medicine, 131(1), 20 (2016).

[2] N. Dhanachandra, K. Manglem & Chanu, **"*Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm*",** Procedia Computer Science, 54 (1), 764 (2015).

[3] M. Nilashi, O. bin Ibrahim, N. Ithnin, & N. H. Sarmin,**" *A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS*",** Electronic Commerce Research and Applications, 14(6), 542 (2015).

[4] S. Wang, L. Jiang, & C. Li, **"*Adapting naive Bayes tree for text classification*",** Knowledge and Information Systems, 44(1), 77 (2015).

[5] Y. Zheng, B. Jeon, D. Xu, Q. M. Wu, & H. Zhang, **"*Image segmentation by generalized hierarchical fuzzy C-means algorithm*",** Journal of Intelligent & Fuzzy Systems, 28(2), 961 (2015).

[6] J. L. Puga, M. Krzywinski, & N. Altman, **" *Points of significance: Bayes theorem*"**, Nature Methods, 12 (4), 277 (2015).

[7] Dai, W., & Ji, W. **"*A MapReduce implementation of C4.5 decision tree algorithms*",** International journal of database theory and application, 7(1), 49 (2014).

[8] T. Menzies, J. Greenwald, and A. Frank, **"*Data Mining Static Code Attributes to Learn Defect Predictors*"**, IEEE Transaction Software Enginering, 33(1), 2 (2007).

[9] Brian Fitzgerald, **"*The Transformation of Open Source Software*",** Management Information system Quarterly, 30(3), 587 (2006).

[10] J. Padhye, V. Firoiu, and D. Towsley. **"*A stochastic model of TCP Reno congestion avoidance and control Technical Report*"**, 1st Ed., Graduate Research Center Amherst, University of Massachusetts, USA (1999).

[11] G.Q. Kenney, **"*Estimating Defects in Commercial Software during Operational Use*"**, IEEE Transaction Reliability, 42(1), 107 (1993).

[12] Dayana Hernandez, **"*An Experimental Study of K* Algorithm*"**, International Journal of Information Engineering and Electronic Business, 2(1), 14 (2015).

[13] Qinbao Song, Zihan Jia, Martin Shepperd, Shi Ying, and Jin Liu, **" *A General Software Defect-Proneness Prediction Framework*",** IEEE Transactions On Software Engineering, 37(3), 356 (2011).

[14] Trung T. Dinh-Trong and James M. Bieman. **"*A Replication Case Study of Open Source Development*",** The FreeBSD Project, IEEE Transactions on Software Engineering, 31(6), 481 (2005).

[15] P. Runeson, C. Andersson, T. Thelin, A. Andrews, and T. Berling, **"*What Do We Know about Defect Detection Methods*",** IEEE Transactions on Software Engineering, 23(3), 82 (2006).

[16] S. and M. Muthulakshmi, **"*Comparative Analysis of Bayes and Lazy Classification Algorithms*"**, International Journal of Advanced Research in Computer and Communication Engineering, 2(8), 3118 (2013).

[17] Michael Goulde and Eric Brown, **"*Open Source Software: A Primer for Health Care Leaders*",** 3rd International Workshop on Predictor Models in Software Engineering, USA, Conference 10, 1109 (2007).

[18] Lan H. Witten and Epik Frank, **"*Data Mining: Practical Machine Learning tools and techniqueswith Java Implementations*"**, Association Computing Machinery, 31(1), 338 (2002).

[19] Lawrence A. Birnbaum, **"*Machine Learning Proceedings*"**, 10th Ed., Morgan Kuafman Publishers, USA (1993).