

Fuzzy Unordered Rule Induction Algorithm Application in Basic Programming Language Competence: A Rule-Based Model

Las Johansen B. Caluza

Information Technology and Computer Education Department, Leyte Normal University, Paterno St., Downtown, Tacloban City, 6500 Leyte, Philippines; lasjohansencaluza@lnu.edu.ph

Abstract

Objective: A rule-based model using Fuzzy Unordered Rule Induction Algorithm (FURIA) aimed to be developed in this study.

Methods/Statistical Analysis: This study utilized a Machine-Learning algorithm called FURIA in developing the model. Graduates of the K to 12 programs in the Philippines who enrolled in the Bachelor of Science in Information Technology in a State University were the respondents of this study. Respondents were assessed using a self-made questionnaire in basic programming using Java programming language and validated both face validation and the internal reliability test with a Cronbach alpha value of 83. Furthermore, the steps performed in this study were the variable definition and data collection, data processing and cleaning, machine-learning algorithm application, verifying and interpreting the model, and reporting the model and synthesizing. **Findings:** This study revealed seven (7) rule-based models with a substantial rating in Kappa statistics and a very high accuracy rating in predicting the competencies by class. **Applications/ Improvements:** The researcher believed the need to unveil the competence of the Senior High School graduates in the K to 12 programs of the Department of Education in the Philippines in basic programming, which aimed to retool the type of teaching strategies to fit the competencies of the student.

Keywords: Basic Programming, Competence, Data Mining, Fuzzy Algorithm, Fuzzy Unordered Rule Induction Algorithm (FURIA), Machine Learning, Rule Model

1. Introduction

The basic principle in the logic formulation for developing skills in programming is an indispensable and integral function in developing analytical skills for complex problem-solving. Arguments relative to the best approach in teaching programming like direct exposure or taking up with basic programming¹ are very evident particularly when the school adopted the Outcomes-Based Education (OBE). Mathematics is not the only prerequisite of programming subjects, but also an adequate command in English is essential². Anent to this, many literatures claim some cognitive issues on learning computer programming such as the definition of computer programming, levels of programming skill development, the demands

of learning to program, and programming as a cognitive activity³. This gap showed a significant impact on the performance of the student in ICT and in the programming course itself.

On the contrary, predicting student performance has been in the limelight for many years in understanding and unveiling the root of how we can help improve the performance of the students and the graduates itself. A study in predicting self-efficacy in test preparation where gender, value, anxiety, test performance, and strategies were conducted utilizing hierarchical multiple regression where the ability in predicting the independent variables in the first model showed $R^2 = 0.3$, second model was $R^2 = 0.211$, and the third model was $R^2 = 0.108$ ⁴. Another study utilized the same approach in predicting the quality

of life after the graduation of senior college students with disabilities which were the stepwise multiple regression analyses⁵. In this research, R-squared were used to determine how close the data are to the fitted regression line, where R-squared is always between 0 and 100%, however, to determine the goodness of fit, it is also essential to look into the residual plots⁶.

In the Philippines, it is very evident that the effect of educational reform made and mandated by the government was the education arm joined together in implementing the Philippine Qualifications Framework (PQF). PQF describes the levels of educational qualifications and sets the standards for qualification outcomes as Senator Villanueva authored and sponsored RA 10968⁷. The Philippine qualification framework⁸ assured quality graduates that can add the human resource of the country. The involved agencies in implementing this framework were the Department of Education (DepEd) for primary education, Technical Education and Skills Development (TESDA), and the Commission on Higher Education (CHED) for Baccalaureate to Post-Doctoral programs. Recently, Senior High School Graduates (SHSG) in the Philippines were accepted to enroll regardless of their tracks or specialization in their senior high school on the degree of their choice since it was not stipulated in the recent CHED Memorandum Order 34, series of 2016. In line with this, competencies relative to the field of specialization they have chosen remain unknown since they can choose any bachelor's degree regardless of their tracks in Senior High School.

Based on the different kinds of literature mentioned in the previous discussions, it is very evident that prediction studies merely focused on varied topics and subjects that were not in line with the current study.

In general, predicting SHS graduate competencies in basic programming remain unexplored. Therefore, it is empirical to investigate this emerging phenomenon in order to design a model based on Fuzzy Unordered Rule Induction Algorithm (FURIA).

2. Learned Literature

2.1 The Rule-Based Algorithm

The rule-based machine learning algorithm is widely used in computer science where the goal is to compete with other machine learning procedures that identify,

learns, or evolves 'rules' to store, manipulate or apply^{9,10}. This type of algorithm includes artificial immune systems, association rule learning, and learning classifier systems and other algorithms that rely on the set of rules where each is covering conceptual and contextual knowledge. Rules typically take in the form of {IF:THEN} expression for example {IF <condition> THEN <result>}

Many researchers had made significant innovation of the rule-based algorithm such as a trainable rule-based algorithm for word segmentation. The algorithm was useful both as a high-accuracy stand-alone segmenter and as a postprocessor that improves the output of existing word segmentation algorithms¹¹. Another study utilized the rule-based algorithm for assigning myocardial fiber orientation to computational heart models. The developed model is called the Laplace-Dirichlet Rule-Based (LDRB) algorithm for assigning myocardial fiber orientation to computational heart models with speed, accuracy, and high usability¹². Recently, an evaluation study between rule-based algorithm and feature selection approaches for classification was made. In the study, it utilized the rule-based algorithm called Prism, which works with the concept of target class and is capable of selecting attributes based on their importance to a particular class¹³.

2.1.1 The RIPPER Algorithm

The RIPPER is a rule-based learning algorithm for classification problems because of its simplicity and compact^{14,15}. Also, RIPPER was able to successfully deal with overfitting^{16,17} and is an excellent choice for learning a fuzzy system where each rule can lessen the impact of a significant problem in fuzzy systems known as the curse of dimensionality¹⁷. RIPPER algorithm was developed for classification problems to generate a crisp rule¹⁵, to overcome this problem, continuous variable needs to be discretized, and the type of discretization can affect the quality of the rules generated by the RIPPER^{18,19}.

Several studies were conducted utilizing the RIPPER algorithm. Repeated Incremental Pruning to Produce Error Reduction or RIPPER is a proportional rule learner; it extracts the rules directly from the data where it works in four stages such as growth, pruning, optimization, and selection¹⁴. A study on cardiac arrhythmia classification using machine learning²⁰ showed how the RIPPER perform and was measure through accuracy surpassing other machine learning algorithms. Also, a study in predicting the severity of closed source of NASA defect reports is

done utilizing RIPPER^{21,22} showed high accuracy rating in predicting the target variable.

2.1.2 The FURIA Algorithm

Fuzzy Unordered Rule Induction Algorithm or FURIA is an extended version of a RIPPER developed by¹⁷. FURIA learns fuzzy rules instead of conventional rules and unordered rulesets instead of rule lists. Fuzzy rules are more general than conventional rules and have some advantages while, most conventional rule learners is a decision list. Below is an example of the antecedent fuzzification algorithm for a single rule r in Algorithm 1 developed by¹⁷.

Algorithm 1 The antecedent fuzzification algorithm for a single rule r

```

1: Let  $A$  be the set of numeric antecedents of  $r$ 
2: while  $A \neq \emptyset$  do
3:    $a_{\max} \leftarrow \text{null}$  { $a_{\max}$  denotes the antecedent with the highest purity}
4:    $\text{pur}_{\max} \leftarrow 0$  { $\text{pur}_{\max}$  is the highest purity value, so far}
5:   for  $i \leftarrow 1$  to  $\text{size}(A)$  do
6:     compute the best fuzzification of  $A[i]$  in terms of purity
7:      $\text{pur}_{A[i]} \leftarrow$  be the purity of this best fuzzification
8:     if  $\text{pur}_{A[i]} > \text{pur}_{\max}$  then
9:        $\text{pur}_{\max} \leftarrow \text{pur}_{A[i]}$ 
10:       $a_{\max} \leftarrow A[i]$ 
11:     end if
12:   end for
13:    $A \leftarrow A \setminus a_{\max}$ 
14:   Update  $r$  with  $a_{\max}$ 
15: end while

```

For the fuzzification of a single antecedent ($A_i \in I_i$) it is important to consider only the relevant training data D_T^i , i.e., to ignore those instances that are excluded by any other antecedent ($A_j \in I_j^F$), $j \neq i$:

$$D_T^i = \left\{ x = (x_1 \dots x_k) \in D_T \mid I_j^F(x_j) = 0 \text{ for all } j \neq i \right\} \subseteq D_T \quad (1)$$

We partition D_T^i into the subset of positive instances, D_{T+}^i , and negative instances, D_{T-}^i . To measure the quality of a fuzzification, the rule purity will be used:

$$\text{pur} = \frac{P_i}{P_i + n_i} \quad (3)$$

Where

$$P_i \stackrel{\text{def}}{=} \sum_{x \in D_{T+}^i} \mu_{A_i}(x)$$

$$n_i \stackrel{\text{def}}{=} \sum_{x \in D_{T-}^i} \mu_{A_i}(x)$$

3. Model Development Methodology

The study utilized Machine Learning (ML) process to derive a model and used to predict the competence of the graduates of Senior High School (SHS) enrolled in a State University in the Philippines in basic programming using Java programming language as a tool. ML algorithms build a mathematical model of sample information, experienced as training data to make predictions or decisions without being explicitly programmed to perform the task²³. The processes performed in this study are shown.

3.1 Variable Definition and Data Collection

The variables used in this study are based on the personal information and perceived competence on the different generic topics in basic programming of the respondents. Table 1 shows the variable definition in this study. The instrument used in this study is a self-made questionnaire and was validated both face validation and reliability test. In the face validation, the research asked an English language expert to check, revised the grammar and its reading understandability with gunning fog index value of 13, and considered a college freshman level²⁴ while the internal consistency test was conducted after. Here were 30 freshmen student who have the same characteristics of the respondents was utilized to answer the questionnaire and measured its internal consistency with a Cronbach alpha value of 0.83 and considered better or accepted in a Likert Scale questionnaire. In the instrument, the competencies have specific descriptions describing the different topics such as basic programming, operators, loop, and arrays. Each description, the respondents will assess their knowledge of the different topics.

3.2 Graphical Presentation of All Attributes

This study utilized convenience sampling on the basis where the members of the sample are conveniently accessible. The instrument was personally distributed and retrieved by the researcher to the 70 freshmen students enrolled in Bachelor of Science in Information Technology in a Philippine State University.

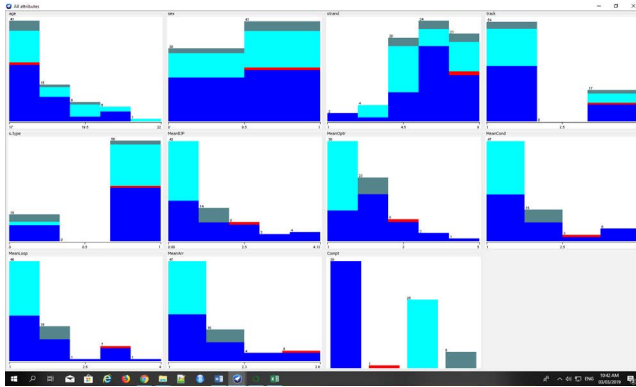


Table 1. Variable definitions

Variable	Value	Description
Age	Scale	Numerical
Sex	0,1	0-Female, 1-Male
Strand	1 to 8	1-ABM, 2-CHS, 3-CSS, 4-GAS, 5-HUMSS, 6-ICT, 7-STEM, 8-OldCur
Track	1 to 4	1-ACAD, 2-TVL,3-SPORTS, 4-Old
School Type	0,1	0-Private, 1-public
Basic	1 to 5	1-No Knowledge at All, 2-Newbie, 3-Intermedia, 4-Advance, 5-Expert
Operators	1 to 5	1-No Knowledge at All, 2-Newbie, 3-Intermedia, 4-Advance, 5-Expert
Conditional Statements	1 to 5	1-No Knowledge at All, 2-Newbie, 3-Intermedia,4-Advance, 5-Expert
Looping Statements	1 to 5	1-No Knowledge at All, 2-Newbie,3-Intermedia,4-Advance, 5-Expert
Array	1 to 5	1-No Knowledge at All, 2-Newbie, 3-Intermedia,4-Advance, 5-Expert
Competence Level	1 to 5	1-No Knowledge at All, 2-Newbie, 3-Intermedia, 4-Advance, 5-Expert

3.3 Data Preprocessing and Cleaning

The datasets collected in this study was encoded using MS Excel and saved with XLSX extension name. Removal of information that was inconsistent and incomplete was performed to maintain its consistency and reliability of datasets. The refined datasets are now converted into Comma Separated Value or CSV format; this is because

the version of the R programming software accepts only this kind of format. This process in data mining technique that involves the transformation of raw data into understandable format and removal of incomplete, inconsistent, and lacking in certain behaviors or trends, and is likely to contain errors is called data preprocessing and cleaning^{25,26}.

3.4 Machine Learning Algorithm Application

This study utilized the Weka application version 3.8.3 in applying the FURIA to 70 datasets. FURIA was used in generating rules in predicting programming competence of senior high school graduates enrolled in Programming 1 using Java Programming language as a basis for developing teaching strategies fit to their level of competence. Fuzzy rules are more general than conventional rules and have some advantages, such as FURIA learns fuzzy rules instead of conventional rules; it makes use of an efficient rule stretching method¹⁷.

3.5 Verifying and Interpreting the Model

10-folds Cross-validation was applied in this study to evaluate and test the generated model. Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and test set to evaluate it²⁷. In *k-fold* cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples use as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation²⁸. The benefit of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once²⁹.

The model was explained through the following:

True Positives (TP) Rate – number of examples predicted that are actually positive.

False Positives (FP) Rate – number of examples predicted positive that isactually negative.

True Negatives (TN) Rate – number of examples predicted negative that isactually negative.

False Negatives (FN) Rate – number of examples predicted negative that are actually positive.

Confusion Matrix with a positive class:

	a	b	< - - classified as
actual a = 0	TP	FN	
actual b = 1	FP	TN	

Confusion Matrix with a negative class:

	a	b	< - - classified as
actual a = 0	TN	FP	
actual b = 1	FN	TP	

Recall (sensitivity): Proportion of instances classified as a given class divided by the actual total in that class (equivalent to TP Rate).

$$Recall = \frac{tp}{tp + fn}$$

Precision Proportion of instances that are truly of a class divided by the total instances classified as that class.

Precision is also known as Positive Predictive Value (PPV).

$$Precision = \frac{tp}{tp + fp}$$

Receiver Operating Characteristics (ROC) Area measurement: The value gives an idea of how the classifiers are performing in general the higher the ROC, the better (Between 0 and 1).

The perfect test (Figure 1) will have a ROC curve that passes through the upper left corner (corresponding to 100 % sensitivity and 100 % specificity). Generally, the closer a ROC curve is to the upper left corner, the better the test is³⁰.

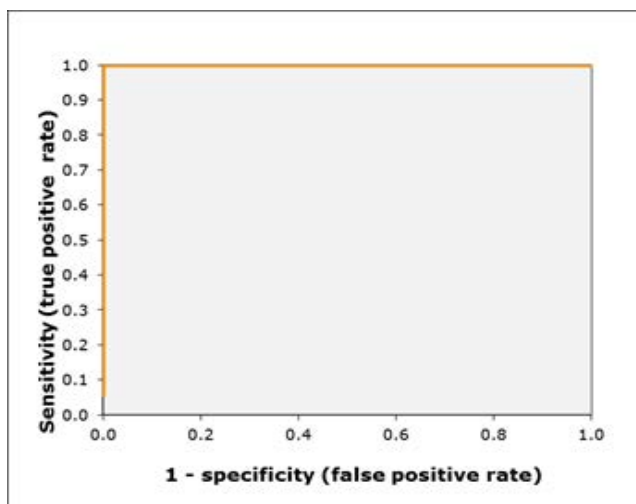


Figure 1. Receiver operating curve.

PRC Area: The precision-recall curve is a Plot of Precision Versus (PPV) recall (sensitivity) for all potential cut-offs for a test.

The perfect test will have a PRC that passes through the upper right corner (corresponding to 100% precision and 100% recall), Figure 2. Generally, the closer a PRC is to the upper right corner; the better the test is³⁰.

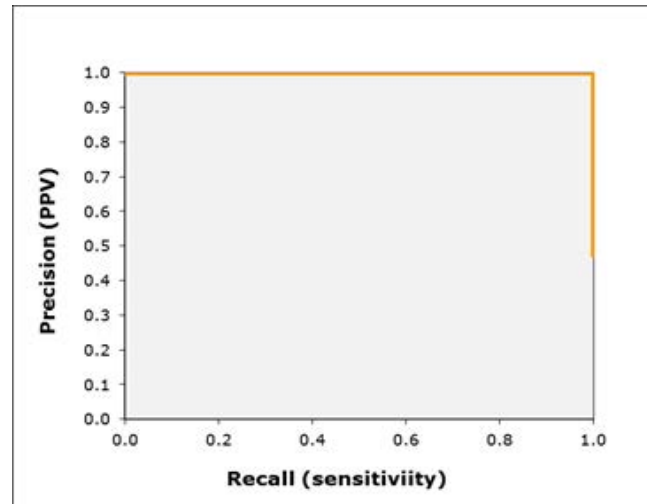


Figure 2. The precision-recall curve.

The reliability of the model was measured using the Kappa index (Kappa statistics) as shown in Table 2³¹.

Table 2. The range of the kappa index and the corresponding agreement between the model and reality³²

Kappa Index	Agreement
0.80 – 1	Almost Perfect
0.60 – 0.80	Substantial
0.40 – 0.60	Moderate
0.20 – 0.40	Fair
0 – 0.20	Slight
≤ 0	Poor

3.6 Reporting the Model and Synthesizing

An in-depth analysis of the model showing the importance and the implication towards the real-time application of the generated model is presented in this segment. The analysis of the result is supported by known literature showing how the gaps were answered in this study.

4. Results and Discussion

The model generated in FURIA shown below consists of seven (7) rules in predicting basic programming skills using Java programming language. Rule 1 show that the basic java programming skills include *number systems concepts, java program structure, variables, data types, identifiers, I/O syntax, parsing, and methods* were among the most significant predictors together with knowledge in Array. Arrays application were determined through the *single dimensional array, copying arrays, searching, sorting, and array class application*. Both basic and array java programming skills were strong predictors in predicting programming competence with a rating of “Expert” with a Cumulative Frequency (CF) = 0.96. Another predictor was the knowledge in conditional statements. It measures their understanding in using *if statement, if else statement, if else if statement, nested statement, switch statement and the use of break statement* with the basic java programming skills resulting to a rating of “Expert” with CF = 0.94. Knowledge in operators were also determined using their understanding in *assignment operators, mathematical operators, and logical operators* with a CF = 0.94. Moreover, looping statements knowledge and skills were also measure through its perceptions in *for loop statement, while loop statement, and do while loop statement* with a negative infinity (-inf) value of conditional statements would result to “Expert” with CF = 0.87. Meanwhile, a student who graduated senior high school with a Strand of Computer Hardware Servicing (CHS) and Computer Systems Servicing (CSS) as perceived by the respondents with a -inf in basic java programming skills, conditional statements, and operators would result to a No Knowledge at All (NKALL) with a CF = 0.94. Finally, if the perceived competence of the respondents in conditional statements and basic java programming skills is positive +inf but their knowledge in operators with basic java programming skills is negative +inf would result to “Newbie” with a CF = 0.77.

These skills are required and prerequisites to more advanced and complex problem-solving in the real world situation. In the implementation of the K to 12 programs in the Philippines, the Department of Education enforces the different schools including private schools to operationalized its offerings in the Senior High School Programs that to include ICT Track.

The government envisioned to have a more competent human resource in the future through its graduates. This

implementation is supported by the different agencies that to include the Technical Education and Skills Development Authority (TESDA), Commission on Higher Education (CHED), Department of Labor and Employment (DOLE), Professional Regulation Commission (PRC), and others. This implementation is through the Republic Act No. 105533, also known as the *Enhanced Basic Education Act of 2013*^{32,33}. However, despite the effort of the national government in the implementation of the K to 12 programs, many still lack in materials and facilities³⁴. Hence, the implementation of digital services requires problem-solving, design skills, logical thinking, understanding how computers and networks operate, and most importantly programming competence³⁵.

Table 3. Summary

Correctly Classified Instances	61	87.14%
Incorrectly Classified Instances	9	12.86%
Kappa statistic	0.7756	
Mean absolute error	0.0823	
Root mean squared error	0.2582	
Relative absolute error	22.00%	
Root relative squared error	59.89%	

FURIA rules:

1. (MeanBJP in [1, 1.125, inf, inf]) and (MeanArr in [-inf, -inf, 1.8, 2]) => Compt=Expert (CF = 0.96)
2. (MeanCond in [1, 1.2, inf, inf]) and (MeanBJP in [1.875, 2, inf, inf]) => Compt=Expert (CF = 0.94)
3. (MeanCond in [1, 1.2, inf, inf]) and (MeanBJP in [-inf, -inf, 1.375, 1.75]) => Compt=Expert (CF = 0.93)
4. (MeanOptr in [2, 2.666667, inf, inf]) => Compt=Expert (CF = 0.94)
5. (MeanLoop in [1, 2, inf, inf]) and (MeanCond in [-inf, -inf, 1.4, 2]) => Compt=Expert (CF = 0.87)
6. (MeanBJP in [-inf, -inf, 1, 1.125]) and (MeanCond in [-inf, -inf, 1, 1.2]) and (strand in [2, 3, inf, inf]) and (MeanOptr in [-inf, -inf, 1, 3]) => Compt=NKALL (CF = 0.95)
7. (MeanCond in [1.8, 2, inf, inf]) and (MeanBJP in [-inf, -inf, 1.875, 2]) and (MeanBJP in [1.375, 1.75, inf, inf]) and (MeanOptr in [-inf, -inf, 2, 2.666667]) => Compt=Newbie (CF = 0.77)

Number of Rules: 7

FURIA generated rule model showed a strong prediction rate with 87.14% correct classification and 12.86%

Table 4. Detailed accuracy by class

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.821	0.065	0.941	0.821	0.877	0.751	0.925	0.937	Expert
	1.000	0.089	0.862	1.000	0.926	0.886	0.967	0.893	NKALL
	0.667	0.047	0.571	0.667	0.615	0.578	0.969	0.637	Newbie
Weighted Avg.	0.871	0.072	0.881	0.871	0.872	0.785	0.944	0.896	

Table 5. Confusion matrix

a	b	c	<-- classified as
32	4	3	a = Expert
0	25	0	b = NKALL
2	0	4	c = Newbie

Table 6. Predictions in test data

=== Classifier model for fold 1 ===					=== Classifier model for fold 2 ===				
inst#	actual	predicted	error	Predict-tion	inst#	actual	predicted	error	Predict-tion
1	2:NKALL	2:NKALL		1	1	2:NKALL	2:NKALL		1
2	2:NKALL	2:NKALL		1	2	2:NKALL	2:NKALL		1
3	2:NKALL	2:NKALL		1	3	2:NKALL	2:NKALL		1
4	1:Expert	1:Expert		1	4	1:Expert	3:Newbie	+	1
5	1:Expert	1:Expert		1	5	1:Expert	1:Expert		1
6	1:Expert	1:Expert		0.711	6	1:Expert	3:Newbie	+	1
7	1:Expert	1:Expert		1	7	1:Expert	1:Expert		1
=== Classifier model for fold 3 ===					=== Classifier model for fold 4 ===				
inst#	actual	predicted	error	Predict-tion	inst#	actual	predicted	error	Predict-tion
1	2:NKALL	2:NKALL		1	1	2:NKALL	2:NKALL		1
2	2:NKALL	2:NKALL		1	2	2:NKALL	2:NKALL		1
3	2:NKALL	2:NKALL		1	3	2:NKALL	2:NKALL		1
4	1:Expert	2:NKALL	+	0.558	4	1:Expert	1:Expert		1
5	1:Expert	1:Expert		1	5	1:Expert	1:Expert		1
6	1:Expert	1:Expert		1	6	1:Expert	1:Expert		1
7	1:Expert	1:Expert		1	7	1:Expert	1:Expert		1
=== Classifier model for fold 5 ===					=== Classifier model for fold 6 ===				
inst#	actual	predicted	error	Predict-tion	inst#	actual	predicted	error	Predict-tion
1	2:NKALL	2:NKALL		1	1	2:NKALL	2:NKALL		1
2	2:NKALL	2:NKALL		1	2	2:NKALL	2:NKALL		1
3	2:NKALL	2:NKALL		1	3	1:Expert	1:Expert		1

4	1:Expert	1:Expert		1	4	1:Expert	1:Expert		0.546
5	1:Expert	1:Expert		1	5	1:Expert	1:Expert		1
6	1:Expert	1:Expert		1	6	1:Expert	2:NKALL	+	0.517
7	3:Newbie	1:Expert	+	1	7	3:Newbie	3:Newbie		1
=== Classifier model for fold 7 ===					=== Classifier model for fold 8 ===				
inst#	actual	predicted	error	Predict-tion	inst#	actual	predicted	error	Predict-tion
1	2:NKALL	2:NKALL		1	1	2:NKALL	2:NKALL		1
2	2:NKALL	2:NKALL		1	2	2:NKALL	2:NKALL		1
3	1:Expert	1:Expert		1	3	1:Expert	2:NKALL	+	1
4	1:Expert	1:Expert		1	4	1:Expert	2:NKALL	+	1
5	1:Expert	1:Expert		1	5	1:Expert	1:Expert		1
6	1:Expert	1:Expert		0.501	6	1:Expert	1:Expert		1
7	3:Newbie	3:Newbie		1	7	3:Newbie	3:Newbie		1
=== Classifier model for fold 9 ===					=== Classifier model for fold 10 ===				
inst#	actual	predicted	error	Predict-tion	inst#	actual	predicted	error	Predict-tion
1	2:NKALL	2:NKALL		1	1	2:NKALL	2:NKALL		1
2	2:NKALL	2:NKALL		1	2	2:NKALL	2:NKALL		1
3	1:Expert	1:Expert		1	3	1:Expert	1:Expert		1
4	1:Expert	1:Expert		1	4	1:Expert	1:Expert		1
5	1:Expert	1:Expert		1	5	1:Expert	1:Expert		1
6	1:Expert	3:Newbie	+	0.717	6	1:Expert	1:Expert		1
7	3:Newbie	3:Newbie		1	7	3:Newbie	1:Expert	+	0.566

incorrect classification with a kappa statistics of 0.77 as shown in Table 3. The Kappa statistic (or value) is a metric that compares an Observed Accuracy with an Expected Accuracy (random chance). The kappa statistic³⁶ is used not only to evaluate a single classifier but also to evaluate classifiers amongst themselves. Therefore, the probability of agreeing in predicting the competence in Basic Programming is 77% (good agreement or Substantial) as perceived by the respondents.

The model identified three (3) classes such as Expert, Newbie, and NKALL out five (5) classes. As perceived by the respondents as illustrated in Table 4. The TP rate of class *Expert* and *NKALL* showed a high prediction rate with 82.1% and 100% respectively, while the TP rate for *Newbie* also showed a good prediction rate of with 66.7%.

This is supported by a low FP rate of the three (3) classes with 6.5% (Expert), 8.9% (NKALL), and 4.7% (Newbie). Meanwhile, in terms of precision of the different classes, it also showed an acceptable rating such

as 94.1% (Expert), 86.2% (NKALL), and 57.1 (Newbie). Moreover, the accuracy of the model by classes was found through the ROC curve, where the higher the value, the more beneficial the test solution. In this case, all the classes received a very high accuracy in ROC Area such as 92.5% (Expert), 96.7% (NKALL), and 96.9% (Newbie) resulting to a *Weighted Average* of ROC Area of 94.4%. The model showed its classification score in confusion matrix as shown in Table 5 where:

- 32 instances are correctly classified as *Expert*, and two (2) misclassified instances rated *Expert* when it is a *Newbie*.
- 25 instances are correctly classified as *NKALL*, and four (4) misclassified instances rated as *NKALL* when it is an *Expert*.
- There are four (4) instances that are classified as *Newbie*, and three (3) misclassified instances rated as *Newbie* when it is an *Expert*.

The model generated was tested using 10-folds cross-validation. As suggested by Beleites, Neugebauer, and Bocklitz³⁷⁻⁴², a small microarray dataset with at least 53 to 208 or any considered as small sample size, re-sampling strategies like bootstrap is repeated or iterated *k-fold* cross validation are the most appropriate tool to sample test. In this case, a Table 6 show how the model was tested where actual value had its corresponding predicted value and was marked by (+) if an error prediction occurs.

5. Conclusion

The use of FURIA in developing a model, predicting the competencies of Senior High School (SHS) Graduates in a State University in basic programming using Java computer language was developed. The generated model was tested and showed a substantial result in the rates and very high prediction rate or accuracy rate by class. This revelation would help the teachers in devising their teaching strategies and teaching styles based on the capabilities of the learners.

Further research maybe conducted on the learning and teaching styles of these SHS graduates to align their competencies, type of teaching styles and learning styles. Moreover, this model can be used to develop a system that will determine the competencies of the SHS graduates in basic programming using the Java programming language.

6. References

- Burton PJ, Bruhn RE. Teaching programming in the OOP era, ACM SIGCSE Bulletin. 2003; 35(2):111–14. <https://doi.org/10.1145/782941.782993>.
- Structured programming in Java. Date accessed: 2002. <https://pdfs.semanticscholar.org/9425/26e2feecdf8bc4f7b865c6f32bb0d99052f3.pdf>.
- On the Cognitive Prerequisites of Learning Computer Programming. Date accessed: 1983. https://web.stanford.edu/~roypea/RoyPDF%20folder/A17_Pea_Kurland_83.pdf.
- Predicting self-efficacy in test preparation: Gender, value, anxiety, test performance, and strategies. Date accessed: 13.03.2018. <https://www.tandfonline.com/doi/abs/10.1080/00220671.2018.1437530>.
- Chao PC. Using self-determination of senior college students with disabilities to predict their quality of life one year after graduation, European Journal of Educational Research. 2018; 7(1):1–8. <https://doi.org/10.12973/euler.7.1.1>.
- Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit. Date accessed: 30.02.2013. <http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit>.
- Villanueva hails signing of law institutionalizing Philippine Qualifications Framework. Date accessed: 22.01.2018. <https://ptvnews.ph/villanueva-hails-signing-law-institutionalizing-philippine-qualifications-framework/>.
- Philippine Qualifications Framework (PQF). Date accessed: 2018. https://www.ceap.org.ph/upload/download/20138/2723637531_1.pdf.
- Bassel GW, Glaab E, Marquez J, Holdsworth MJ, Bacardit J. Functional network construction in Arabidopsis using rule-based machine learning on large-scale data sets, The Plant Cell. 2011; 23(9):3101–13. <https://doi.org/10.1105/tpc.111.088153>. PMID: 21896882, PMCID: PMC3203449.
- Weiss SM, Indurkha N. Rule-based machine learning methods for functional prediction, Journal of Artificial Intelligence Research. 1995; 3:383–403. <https://doi.org/10.1613/jair.199>.
- A trainable rule-based algorithm for word segmentation. Date accessed: 1996. <https://www.aclweb.org/anthology/P97-1041>.
- Bayer JD, Blake RC, Plank G, Trayanova NA. A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models, Annals of Biomedical Engineering. 2012; 40(10):2243–54. <https://doi.org/10.1007/s10439-012-0593-5>. PMID: 22648575, PMCID: PMC3518842.
- Evaluation of Rule-Based Learning and Feature Selection Approaches For Classification. Date accessed: 09/2018. https://www.researchgate.net/publication/327096660_Evaluation_of_Rule-Based_Learning_and_Feature_Selection_Approaches_For_Classification.
- Fast effective rule induction. Date accessed: 1995. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.2612&rep=rep1&type=pdf>.
- Asadi S. Evolutionary fuzzification of RIPPER for regression: Case study of stock prediction, Neurocomputing. 2019; 331:121–37. <https://doi.org/10.1016/j.neucom.2018.11.052>.
- Foundations of rule learning. Date accessed: 2014. <https://dl.acm.org/citation.cfm?id=2788240>.
- Hühn J, Hüllermeier E. FURIA: An algorithm for unordered fuzzy rule induction, Data Mining and Knowledge Discovery. 2009; 19(3):293–319. <https://doi.org/10.1007/s10618-009-0131-8>.

18. The influence of class order on rule induction results. Date accessed: 03.04.2014. <https://ieeexplore.ieee.org/abstract/document/6841946>.
19. Garcia S, Luengo J, Sáez JA, Lopez V, Herrera F. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning, *IEEE Transactions on Knowledge and Data Engineering*. 2013; 25(4):734–50. <https://doi.org/10.1109/TKDE.2012.35>.
20. Cardiac Arrhythmia Classification Using Machine Learning Techniques. Date accessed: 31.10.2018. https://link.springer.com/chapter/10.1007/978-981-13-1642-5_42.
21. Comparing mining algorithms for predicting the severity of a reported bug. Date accessed: 04.03.2011. <https://ieeexplore.ieee.org/document/5741332>.
22. Predicting the Severity of Closed Source Bug Reports Using Ensemble Methods. Date accessed: 01/2019. https://www.researchgate.net/publication/328745999_Predicting_the_Severity_of_Closed_Source_Bug_Reports_Using_Ensemble_Methods_Proceedings_of_the_Second_International_Conference_on_SCI_2018_Volume_2.
23. Christopher M. Bishop. Pattern recognition and machine learning (Information Science and Statistics). Qatar International Zebrafish Workshop: Theoretical, Practical Lectures by Renowned Experts Qatar University; Springer 2010. p. 738. https://www.goodreads.com/book/show/55881.Pattern_Recognition_and_Machine_Learning.
24. Bogert J. In Defense of the Fog Index, *Business Communication Quarterly*. 1985; 48(2):9-12.
25. Mohit Sharma. What Steps should one take while doing Data Preprocessing? <https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa>.
26. Hand DJ. Principles of data mining, *Drug Safety*. 2007; 30(7):621–22. <https://doi.org/10.2165/00002018-200730070-00010>. PMID:17604416.
27. 10-fold Cross-validation. Date accessed: 2019. <https://www.openml.org/a/estimation-procedures/1>.
28. Fast Cross-Validation for Kernel-based Algorithms. Date accessed: 14.01.2019. <https://www.ncbi.nlm.nih.gov/pubmed/30640598>.
29. A study of cross-validation and bootstrap for accuracy estimation and model selection. Date accessed: 1995. <http://ai.stanford.edu/~ronnyk/accEst.pdf>.
30. Precision-recall curves – what are they and how are they used? Date accessed: 03/2017. <https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used>.
31. Guzzetti F, Reichenbach P, Ardizzone F, Cardinali M, Galli M. Estimating the quality of landslide susceptibility models, *Geomorphology*. 2006; 81(1-2):166–84. <https://doi.org/10.1016/j.geomorph.2006.04.007>.
32. Cohen J. A coefficient of agreement for nominal scales, *Educational and Psychological Measurement*. 1960; 20(1):37–46. <https://doi.org/10.1177/001316446002000104>.
33. Republic Act No. 10533. Date accessed: 15.05.2013. <https://www.officialgazette.gov.ph/2013/05/15/republic-act-no-10533/>.
34. Lack of materials, facilities still hound K to 12 implementation. Date accessed: 02/06/2014. <https://www.gmanetwork.com/news/news/specialreports/363734/lack-of-materials-facilities-still-hound-k-to-12-implementation/story/>.
35. Tuomi P, Multisilta J, Saarikoski P, Suominen J. Coding skills as a success factor for a society, *Education and Information Technologies*. 2018; 23(1):419–34. <https://doi.org/10.1007/s10639-017-9611-4>.
36. McHugh ML. Interrater reliability: the kappa statistic, *Biochemia Medica: Biochemia Medica*. 2012; 22(3):276–82. <https://doi.org/10.11613/BM.2012.031>.
37. Beleites C, Neugebauer U, Bocklitz T. Sample size planning for classification models, *Analytica Chimica Acta*. 2013; 760:25–33. <https://doi.org/10.1016/j.aca.2012.11.007>. PMID: 23265730.
38. Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. Date accessed: 18/05/2012. <https://onlinelibrary.wiley.com/doi/10.1111/j.1600-0587.2012.07348.x>.
39. Multicollinearity in Regression Analysis: Problems, Detection, and Solutions. Date accessed: 2019. <https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/>.
40. Assaf GA, Tsionas M, Tasiopoulos A. Diagnosing and correcting the effects of multicollinearity: Bayesian implications of ridge regression, *Tourism Management*. 2019; 71:1–8. <https://doi.org/10.1016/j.tourman.2018.09.008>.
41. George G. Judge, R. Carter Hill, William E. Griffiths, Helmut Ltkepohl, Tsoung-Chao Lee. Introduction to the Theory and Practice of Econometrics. . 2nd Edition. 1988. p. 1064 ISBN-13: 978-0471624141. <https://www.amazon.com/Introduction-Theory-Practice-Econometrics-2nd/dp/0471624144>.
42. Ragothaman S, Naik B, Ramakrishnan K. Predicting corporate acquisitions: An application of uncertain reasoning using rule induction, *Information Systems Frontiers*. 2003; 5(4):401–12.