International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 4-2
ISSN: 2088-5334

Machine Learning Approach for Bottom 40 Percent Households (B40) Poverty Classification

Nor Samsiah Sani[#], Mariah Abdul Rahman[#], Azuraliza Abu Bakar[#], Shahnorbanun Sahran[#], Hafiz Mohd Sarim[#]

[#]*Center For Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia*

E-mail: norsamsiahsani@ukm.edu.my, mariahabdulrahman.ukm@gmail.com, azuraliza@ukm.edu.my, shahnorbanun@ukm.edu.my, hms@ukm.edu.my

***Abstract*— Malaysia citizens are categorised into three different income groups which are the Top 20 Percent (T20), Middle 40 Percent (M40), and Bottom 40 Percent (B40). One of the focus areas in the Eleventh Malaysia Plan (11MP) is to elevate the B40 household group towards the middle-income society. Based on recent studies by the World Bank, Malaysia is expected to enter the high-income economy status no later than the year 2024. Thus, it is essential to clarify the B40 population through a predictive classification as a prerequisite towards developing a comprehensive action plan by the government. This paper is aimed at identifying the best machine learning models using Naive Bayes, Decision Tree and k-Nearest Neighbors algorithm for classifying the B40 population. Several data pre-processing task such as data cleaning, feature engineering, normalisation, feature selection: Correlation Attribute, Information Gain Attribute and Symmetrical Uncertainty Attribute and sampling methods using SMOTE has been conducted to the raw dataset to ensure the quality of the training data. Each classifier is then optimized using different tuning parameter with 10-Fold Cross Validation for achieving the optimal values before the performance of the three classifiers are compared to each other. For the experiments, a dataset from National Poverty Data Bank called eKasih obtained from the Society Wellbeing Department, Implementation Coordination Unit of Prime Minister's Department (ICU JPM), consisting of 99,546 households from 3 different states: Johor, Terengganu and Pahang are used to train each of the machine learning model. The experimental results using 10-Fold Cross-Validation method demonstrates that the overall performance of Decision Tree model outperformed the other models and the significance test specified the result is statistically significance.**

***Keywords*— bottom 40; B40 classification; poverty; decision tree; K-Nearest Neighbors, naïve bayes; parameter tuning.**

I. Introduction

Economists agree that poverty does not have a definite concept [1]. In conventional ecomonics, poverty can be described into four categories, which are monetary approach, capability approach, social exclusion and poverty participatory assessment (PPA) [2]-[4]. The concept and operational of poverty in Malaysia is commonly based on monetary approach perspective. In Malaysia, a poverty threshold known as the Poverty Line Index (PLI) is determined by the Economic Planning Unit (EPU) of the Prime Minister's Department. This threshold obtained data based on income measurement perspective. PLI per capita in Peninsular Malaysia is counted according to peninsular of Malaysia, Sabah and Sarawak. Studies have shown that PLI is larger in city areas compared to non-city and rural regions.

Since the 1970's, the Government of Malaysia has stepping up efforts to eradicate porverty. For instance, there has been a significant reduction in poverty from 49.3% occurences in 1970 to just 3.8% in the year 2009. On the 2009 data, 2.4 million households were identified in the porvery category, with 1.8% of households further identified within the hardcore poor group, 7.6% in the poor group, and the remaining 90.6% in the low-income group. The B40 households had a total household income level of less than RM300 per month while the mean monthly income was RM1, 440 [1].

Following the Tenth Malaysia Plan set in 2011, the government initiated a 4-year plan to improve income levels of those in the B40 households. Households within this group subsequently improved through means of income and capacity building programmes [1]. The government continue to support B40 households through the Eleventh Malaysia Plan set in 2016 by implementing a continued 4-year strategy to raise affected household's income and wealth ownership, addressing the increased cost of living, and strengthening delivery mechanisms [5]. The Household Income and Basic Amenities Survey 2016 report released by the Malaysian Department of Statistics stated that the median income of B40 households has increased to RM3,000 and according to Malaysia Economic Monitor Report in December 2017 by World Bank Malaysia, the threshold value of B40 households increased from RM 3,860 in 2014 to RM 4,360 in 2016 – 2017 [6]. Thus, the determination of B40 households through predictive classification is important to help the government to identify and develop specific actions by engaging further consultations with relevant stakeholders, which include ministries, academia and civil society organisations (CSOs).

In this paper, B40 prediction model were experimented and using three algorithms: Naive Bayes, Decision Tree and k-Nearest Neighbours. This paper comprises of five sections: Section 2 presents some basic information regarding machine learning and reviews several related works on poverty classification worldwide. Section 3 outlay the experimental phase in this study, elaborating on databases used, evaluation methods and statistical analyses adopted for experiment evaluation. The experimental results and discussion are described in Section 4.

Machine Learning is the development of artificially intelligent programs using algorithms to make the programs learn by themselves. This is done by looking at the patterns of certain sets of data without any explicit instruction/rule-based programming [7], [8]. The programs will improve their learning over time based on their experience observing the given data. The prediction goal is to exact an outcome accurately based on reasonable relationships from any provided data. Two main categories that define a majority of learning algorithms methods exist, known as supervised learning and unsupervised learning. Supervised learning is related to a scenario in which the "experience", otherwise referred to as a training example, contains significant information (the labels) missing in the unseen "test examples" of which the learned expertise is to be applied. In unsupervised learning, there is no distinction between training and test data [9].

To date, no prediction task been performed yet to the B40 households. However, there are a variety of prediction methods in AI literature that have been used regarding poverty. A study done by Pareek and Prema [10] used a multi-layer perceptron network to classify the poor in India as Below-Poverty-Line (BPL) and Non-Below-Poverty-Line (Non BPL) using Artificial Neural Network [11], [12].

Other attempts on B40 related prediction include a classification task for poverty in Mauritius using a Decision Tree algorithm. The algorithm is applied to the census data to categorize people based on the relative poverty line. The analysis uncovered several critical variables in the classification of the poverty status of an individual. A byproduct of that study was also on the evidence of a poverty-gender gap in which women have higher chances to be classified as poor in comparison to men.

A Naive Bayes Classifier (NBC) algorithm implemented in [13] to perform classifications of poor families in Indonesia with 11 indicators, uses a total of 219 poor families as the dataset. The experimental results showed that Naive Bayes Classifiers can do classifications of poor families with an accuracy of 93%. They have also done a poverty mapping based on the results obtain from the classification which described the potential of poverty existing in certain regions.

Another study focused on the implementation of a poverty index based on K-Means algorithms. Utilising the poverty index variables, predictive modelling was further studied to predict poverty levels using the Binary logistic regression, Neural networks, Decision trees and Random forests. The study found that Neural networks achieved the best predictive ability compared to the decision trees which is the worst performing algorithm [14].

Most of the studies mentioned above used machine learning method to predict poverty levels using well-known machine learning models: Artificial Neural Network, Naïve Bayes, Logistic Regression, Decision Tree and Random Forest. However, these studies are missing some important concepts in machine learning such as feature selection methods, feature engineering and parameter tuning. Thus, in this paper, a more comprehensive study on those concepts will be conducted.

II. Material and Method

The research methodology of this study is divided into three phases. The dataset phase starts by identifying data examined in this study and analysing its source, details and quantity. This is followed by the Pre-processing phase which aims to prepare the data for processing. Particularly, this phase includes four tasks, which are data cleaning. feature engineering, Normalisation, and sampling method. Pre-processed data was then used in the third phase to establish a comparative analysis among the three techniques to identify the best machine learning technique.

A. Dataset Description

For this study, pre-labeled dataset was used to teach the algorithm to identify B40 group households and for further test on how well it predicts the B40 in especially for unseen data cases. The dataset used in this study comes from National Poverty Data Bank, called 'eKasih', a centralized database which keeps detail profiling of the poor and hardcore poor households in Malaysia. eKasih was developed to assist the government to plan, implement and monitor poverty eradication programs at the national level, and thus, improve the effectiveness of such programs [15], [16]. For this study, a total of 99,546 households records were used from three different states: Johor, Pahang and Terengganu as summarised in Table 1. The eKasih dataset used in this study has 15 attributes and the features are described in Table 2.

TABLE I
Ekasih Dataset

State	Total Households
Johor	23,890
Pahang	22,534
Terengganu	53,122
TOTAL	99,546

TABLE II
ATTRIBUTES FROM EKASIH DATASET

Attribute	Type	Description
State	Nominal	State
Area	Nominal	District
Strata	Nominal	Strata (urban or rural)
Ethnic	Categorical	Ethnic
Marital status	Categorical	Marital status
Age	Continuous	Age
Sex	Nominal	Gender (female or male)
Jobs	Categorical	Occupation
Education	Categorical	Education level
Type of ownership	Categorical	Ownership type of the house
Household number	Discrete	Total number of household
Total income	Continuous	Total income of the household for the past 12 months
Income per capita	Continuous	Per capita income
Date of record	Nominal	Registration date
Poor status	Categorical	Poverty status for the household (Poor, Hardcore Poor, Excluded)

B. *Pre-Processing*

Data pre-processing is a process to transform a dataset so that the information content is best exposed to the mining tool. The data from the real world is always incomplete, inconsistent and may contain noise such as errors and outliers. Thus, data pre-processing is needed to ensure the data is formatted for a given miner tool and needs to be adequate for a given method [17].

In this study, several tasks in data pre-processing such as data cleaning, feature engineering, normalisation, feature selection and sampling methods will be conducted. There are various data mining tools that can be used for data pre-processing purposes. In this study, the 'Waikato Environment for Knowledge Analysis (Weka)' version 3.8 software was used as a tool to perform the pre-processing task. Weka is a java-based machine learning software, developed by the University of Waikato, New Zealand. Weka contains various types of machine learning algorithms and operates on an open source license. It also provides various visualization tools for data analysis and predictive modelling [18].

1) Data Cleaning: Before starting the data cleaning process, data visualization can be utilized to get an overview of the basic pattern of the dataset in a graphical view. Figure 1 shows data visualization for all the attributes in the eKasih dataset.

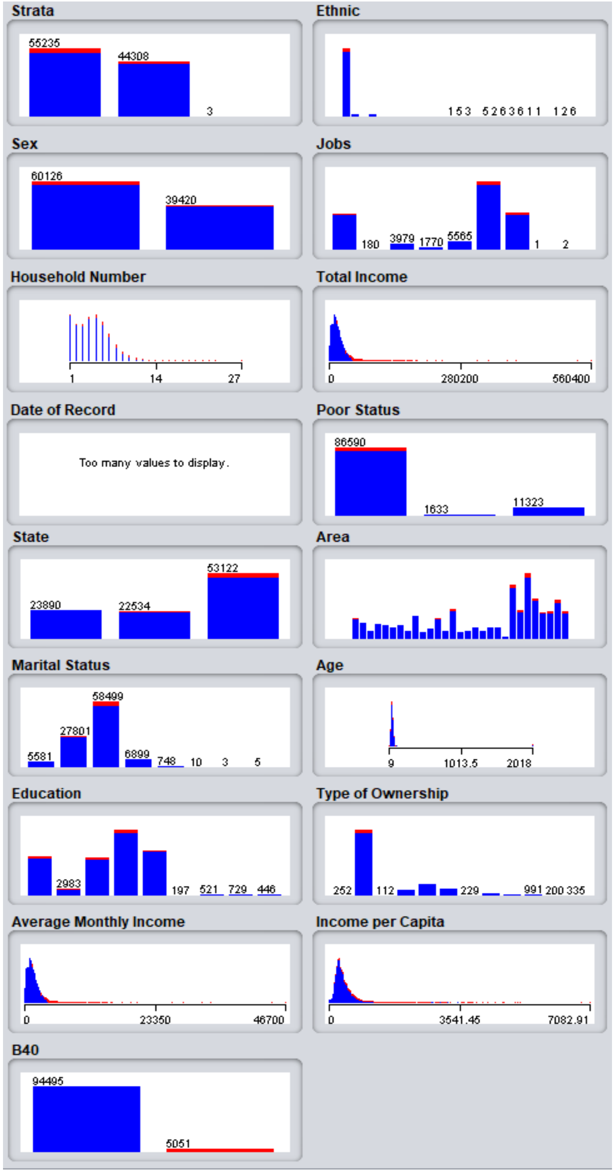

Fig. 1 Data Visualization for Each Attributes.

The dataset contains some missing (null) values based on manual checking done using filtering functions in Weka. The attributes that contain null values are replaced manually as described in Table 3.

TABLE III
ATTRIBUTES CONTAIN MISSING VALUES

Attributes	Total Instances	Replaced Value
Marital status	4	No Information
Education	7	No Education
Type of ownership	213	No Information
Total income	1	0.00
PoorsStatus	1	Hardcore Poor - since the total income is 0

The dataset may also contain outliers. An outlier refers to instances of datasets that deviates from other observations, possibly generated by a different mechanism [19], [20]. In this study, outliers' detection is conducted by using

Interquatile Range in Weka. There are total of 25 outliers detected in the dataset after applying the filter. The outliers are then manually examined using MS Excel. Since the data is a census data, which has been verified by domain experts, therefore the outliers are kept and used in this study.

2) Feature Engineering: The eKasih dataset does not have a B40 category. The B40 threshold is identified based on median monthly income, however this attribute is not available in eKasih. Therefore, the Median Monthly Income attribute is generated using the following formula:

$$\text{Average Monthly Income} = \text{Total Income} / 12 \quad (1)$$

Then, a pre-labelled class for B40 is manually generated based on the following threshold as shown in Table 4.

TABLE IV
DESCRIPTION OF CLASS FEATURE

Class	Description
B40	Johor: Median monthly income < 4830 Pahang: Median monthly income < 3540 Terengganu: Median monthly income < 4070
NOT-B40	Johor: Median monthly income >= 4830 Pahang: Median monthly income >= 3540 Terengganu: median Monthly income >= 4070

3) Normalization: Normalization is a scaling technique based on numeric features, where there is often a large difference between the maximum and minimum values, e.g. 1 and 10000. The normalization will make the value magnitudes scale to appreciably low values ([21], [22]). In this study, normalization is applied the following attributes: Age, Number of Households, Total Income, Average Monthly Income and Per Capita Income, in which their range become from 0 to 1.

4) Feature Selection: Feature Selection is a process to improve classification accuracy by removing irrelevant and redundant features from the original dataset [23]. Feature selection, also known as attributes selection, is used to reduce the dimensionality of the dataset, increase the learning accuracy, and improve result comprehensibility. There are three techniques for Feature Selection: Filter, Wrapper and Embedded [24]. A study done in [25] applied eight feature selection ranking methods to ten different datasets and evaluate half of the top ranked attributes of each ranking method using eight different classifiers to get the classification accuracy. In this study, three ranking methods: Correlation Attribute, Information Gain Attribute and Symmetrical Uncertainty Attributes are evaluated. Section 6 of this paper discusses the experimental results.

5) Sampling Method Selection: In the experimental dataset, the number of minority class (NOT-B40) is dominated by number of majority class (B40) with an imbalance ratio of 5:95. This causes the classifiers to get biased towards the majority class. Two approaches were used to treat this imbalance; 1) data-level approach such as sampling and feature selection and 2) algorithm-level approach such as one class learning, cost sensitive learning and ensemble method ([26], [27]). In this study, an over sampling method called SMOTE [28], which stands for Synthetic Minority Oversampling Technique, is applied to generate synthetic minority examples in order to over-sample the minority class. The dataset has a total number of 99,546 instances, which consists of 94,495 instances for majority class (B40) and 5,051 instances for minority class (NOT-B40). SMOTE is then applied to the dataset at 400% over sampling degree with 5 numbers of nearest neighborhoods, increasing the minority class from 5,051 to 25,255 instances. Table 5 summarizes the number of instances before and after SMOTE is applied.

TABLE V
NUMBER OF INSTANCES BEFORE AND AFTER SAMPLING

	No of Instances	Majority Class	Minority Class
Before Sampling	99,546	94,495 (95%)	5,051 (5%)
After Sampling	119,750	94,495 (79%)	25,255 (21%)

C. Classification Algorithms

1) Naïve Bayes: Naive Bayes algorithm is a simple probabilistic algorithm, applying Bayes' theorem with independence assumptions. Independence here can refer to a naive state, hence the name of this algorithm. Naive Bayes classifiers assume that the presence (or absence) of a feature of a class is unrelated to the presence (or absence) of any other feature. A Naive Bayes algorithm can be trained very efficiently in a supervised learning setting, depending on the precise nature of the probability model. Naive Bayes models uses the method of maximum likelihood for parameter estimation in many practical applications [8, 29].

2) Decision Tree (J48): The decision tree algorithm is made up of three fundamental segments: root node, internal node and leaf node. The root node core starting node, while the leaf node refers to the terminal fundamental of the structure and the internal nodes are the nodes in between the root and the leaf. Internal nodes represent tests on an attribute, while the branch denotes the outcome of the test as each of the leaf node holds a class label. There are various decision tree algorithms available such as Decision Stump, J48, LMT, Random Forest and Random Tree [12, 30]. The J48 classifier, based on the C4.5 algorithm, is used specifically in this study. This algorithm creates a decision tree from on a set of labelled input data which can then be used for classification task [19].

3) k-Nearest Neighbors: The k-Nearest Neighbors algorithm is another simple machine learning algorithms implementation. k-Nearest Neighbors is also known as Memory-Based Classification since the training examples need to be in the memory at run-time ([12], [19]). K-nearest neighbour operates in a way that objects close to each other will exhibit similar characteristics. Therefore, if the characteristic features of one of the objects is discovered, its nearest neighbours can also be predicted. k-Nearest

Neighbors has shown good performance in the classification task of various datasets. The K-Nearest Neighbors algorithm used in this study is called IBk in Weka 3.8

D. Tuning Parameters

Three classification algorithms are compared in this study, which are Naïve Bayes, Decision Tree (J48) and k-Nearest Neighbors (IBk). Each classifier is tuned using different tuning parameters to produce high accuracy results. A series of experiments are conducted to get the optimal values of each classifiers. The performance between the three classifiers are then evaluated and compared.

1) Discretization: For the Naïve Bayes classifier, discretization is considered as a tuning parameter since the dataset used in this study has several continuous attributes. Discretization is the process to transform numeric data into nominal data. This is done by orginazing numeric values into distinct groups, whose length is fixed. It can be employed to estimate its probabilities. Discretization can be done during pre-processing but in this study, discretization is turned on inside the classifier's object editor. The name for discretization parameter in Weka 3.8 is 'useSupervisedDiscretization'.

2) Confidence Factor: In Decision Trees, there are various parameters that can be tuned to increase the classification accuracy. One of the parameters is confidence factor, which determines whether an attribute with a certain value belongs to a certain class [30]. In this study, the J48 classifier is tested with a confidence factor ranging from 0.1 to 1.0 by an increment of 0.2.

3) Minimum Number of Objects: The other tuning parameter for Decision Trees experimented in this study is minimum number of objects which is called 'minNumObj' in Weka 3.8. It specifies the minimum number of instances at the leaf node as a threshold value. It will check the minimum number of object in a leaf whenever the split is made. If the instances at leaf are less than the minimum number of objects specified, the parent node and children node are compressed to a single node [31]. In this study, different ranges of the minimum number of objects are tested for accuracy.

4) k-Value: The k value refers to the k-number of nearest neighbors used in the classification. The k-value is extensively used in k-Nearest Neighbors algorithm. The k parameter was determined using a bootstrap procedure [32]. In this study, the k-value is investigated from 1 to 10 to identify the optimal value for the training samples.

5) Distance Function: In IBk classifier, the distance between two data points in a feature space is measured by distance function [33]. There are four distance functions that are examined in this tuning process; Euclidean distance, Chebyshev distance, Manhattan distance and Minkowski distance. The distance function which gives the best accuracy to the classifier will be selected.

E. Regularization

Inductive learning is a key concept in machine learning, which refers to the process of learning the general concepts from specific examples provided. Specifically, this refers to the attempt to learn target function from available training data. Meanwhile, generalization refers to how well a machine learning model learns the concepts so it can be applied to specific examples that were not seen by the model during the learning. Machine learning's primary objective is to generalize well enough so that the model obtained from the training set can be applied to the unseen data portion from the problem domain. If the algorithm fits the training data too well, it can cause overfitting issues which may lead to poor performance of a machine learning algorithm [34].

Overfitting is an occurrence where the model learns a both the target function and noise during the training, consequently degrading the performance of that model on an unseen data. Overfitting is more likely to happen to nonlinear models that have more flexibility when learning a target function. A number of methods are available to avoid overfitting. For example, a pruning process can be applied to a decision tree model to reduce the size of a tree that are too large and deep [35], and such process was implemented in this study.

Other methods were also applied to overcome overfitting related to specific machine learning models, specifically resampling techniques such as SMOTE oversampling, used to treat imbalanced dataset, and k-fold Cross validation, which iteratively trains and test a model k-times from different training data subsets, to increase generalization chances.

III. Results and Discussion

The classification performance is estimated using Classification Accuracy and Kappa Statistic. Classification Accuracy is often presented as a percentage ratio of the number of correct predictions out of all predictions made, where 100% is the highest an algorithm can achieve. Kappa Statistic refers to the measurement of agreement in prediction for two sets of categorized data. Kappa statistics range is between 0 to 1, in which a lower value indicates a weak argument, while a higher value indicates a strong agreement. Kappa results is interpreted as follows: kappa values less or equal to zero indicate no agreement, kappa values between 0.40 to 0.59 is considered a moderate agreement, 0.6 to 0.79 as substantial and values above 0.8-1.0 indicate as almost perfect agreement [36]. Statistical Test is also conducted in this study to determine whether a classifier's performance is statistically different than another.

A. The Effects of Feature Selection on Classification Accuracies

The experiment of feature selection algorithms on the dataset is conducted using the Correlation Attribute, Information Gain Attribute and Symmetrical Uncertainty Attribute. Table 6 shows the same top eight attributes for all the three ranking methods which are State, Area, Ethnic, Household number, Total income, Average monthly income, Income per capita and Date of record.

TABLE VI
TOP EIGHT ATTRIBUTES FOR FEATURE SELECTION

Feature Selection	Rank	Top 8 Rank Attributes
Correlation Attribute	0.0496537	Total income
	0.0496537	Average monthly income
	0.0033788	Income per capita
	0.0021919	State
	0.0018016	Date of record
	0.0011617	Area
	0.0004468	Ethnic
	0.0003087	Household number
Information Gain Attribute	0.0096794	Date of record
	0.0069222	Total income
	0.0069222	Average monthly income
	0.0052573	Area
	0.0032063	State
	0.0017948	Income per capita
	0.0004822	Household number
	0.0002386	Ethnic
Symmetrical Uncertainty Attribute	0.0417680	Average monthly income
	0.0417680	Total income
	0.0049632	Income per capita
	0.0038750	State
	0.0034789	Date of record
	0.0022288	Area
	0.0006572	Ethnic
	0.0005498	Household number

The experiment is carried out using 10-Fold Cross-Validation test option. Table 7 shows the comparison of classification accuracy for Naïve Bayes, J48 and IBk classifier using all the 16 attributes, compared to only 8 attributes chosen via feature selection. The results clearly indicate that using the top 8 attributes determined and ranked by feature selection improves the classification accuracy on the dataset in which the average accuracy increases from 93.35 % to 95.76 %. Similarly, the Kappa Statistic also shows similar improvement, increasing from 0.82 to 0.87.

TABLE VII
EFFECTS OF FEATURE SELECTION

Classifier	Before Feature Selection (16 attributes)		After Feature Selection (8 attributes)	
	Accuracy (%)	Kappa Statistic	Accuracy (%)	Kappa Statistic
Naïve Bayes	86.80	0.65	91.52	0.75
J48	99.29	0.98	99.18	0.98
IBk	93.97	0.82	96.58	0.90
Average	93.35	0.82	95.76	0.87

B. *The Effects of Tuning Parameters on Classification Accuracies*

1) Naïve Bayes Classifier: Naïve Bayes classifier is tuned using a discretization parameter. Table 8 shows the comparison of classification accuracy before and after utilizing the discretization parameter. The results specified that the accuracy is increased from 91.52 % to 97.27 % after discretization is turned on. Thus, in this study, parameter discretization must be turned on to get the optimal parameter for Naïve Bayes classifier.

TABLE VIII
PARAMETER TUNING RESULTS FOR DISCRETIZATION

Tuning Parameter	Classification Accuracy (%)	Kappa Statistic
Before Discretization	91.52	0.75
After Discretization	97.27	0.92

2) Decision Tree (J48) Classifier: As stated in Section III above, there are two parameter that significantly affect the performance of the J48 classifier, which are the confidence factor and the minimum number of objects. To get the optimal value of the confidence factor, a range of values are tested from 0.1 to 1.0 (by an increment of 0.2) and a minimum number of objects are held at 2. Detailed information about the accuracy obtained for the confidence factor is reported in Table 9. The parameter is tested using 10-Fold Cross Validation.

TABLE IX
PARAMETER TUNING RESULTS FOR CONFIDENCE FACTOR

Confidence Factor	Classification Accuracy (%)
0.2	99.16
0.4	99.27
0.6	99.17
0.8	99.17
1.00	99.17

As shown in Table 9, classification accuracy is increase up to about 0.4 confidence factor at a peak of 99.27% and the accuracy is constant at 99.17% when the confidence factor is above 0.5. Therefore, the optimal value for confidence factor parameter for J48 Classifier is 0.4.

Then, to get the optimal value for a minimum number of objects, a value ranging from 5 and 30 (by the increment of 5) is tested at confidence factor 0.4. The accuracy obtained is reported at Table 10. Experimental results showed the classification accuracy of J48 classifier is decreased when the minimum instance requirement increased. The highest classification accuracy is achieved at 99.27% at the default minimum number of objects. Thus, the optimal value for the minimum number of objects parameter was chosen as 2.

TABLE X
PARAMETER TUNING RESULTS FOR MINIMUM NUMBER OF OBJECTS

Minimum Number of Objects	Classification Accuracy (%)
2 (default)	99.27
5	99.24
10	99.18
15	99.13
20	99.11
25	99.08
30	99.06

3) k-Nearest Neighbors (IBk) Classifier: Key tuning parameter for k-Nearest Neighbors algorithm is the k-value.

In this study, k-value is tested on three different values (1,5 and 10) with four different distance functions (Euclidean Distance, Chebyshev Distance, Manhattan Distance and Minkowski Distance) for choosing the optimal parameter of the IBk classifier.

TABLE XI
TUNING PARAMETER RESULTS FOR K-NEAREST NEIGHBORS

k value	Classification Accuracy (%)			
	Euclidean Distance	Chebyshev Distance	Manhattan Distance	Minkowski Distance
1	96.58	96.09	96.80	96.58
5	95.39	94.07	95.74	95.39
10	94.34	92.66	94.85	94.34

Table 11 shows the result of classification accuracy using 10-Fold Cross-Validation. The results indicate when k value increases from 1 to 10, the classification accuracy decreases, and the results is consistent for all the distance function. The highest classification accuracy (96.80%) is obtained by Manhattan distance functions at k value=1. Therefore, the optimal parameter of the IBk classifier was chosen as k = 1 and distance function is Manhattan distance.

C. *The Performance Evaluation of Different Classifier*

After getting the optimal value of each classifier during the parameter tuning process, the performance between the three selected classification algorithms, which are Naïve Bayes, J48 and IBk, are compared. There are four test options available in Weka 3.8, namely Training Dataset, Supplied Test Set, Percentage Split and Cross Validation. Experiments conducted in this study is measured on 10-Fold Cross Validation.

1) 10-fold Cross-Validation Method: Cross-validation is a statistical method to evaluate predictive models by splitting the original sample into two portions: a training set to learn or train a model, and a test set to perform evaluation on it. For k-fold cross-validation, data are partitioned into k equally sized folds. Training and validations are performed repeatedly for each number of k-iteration. Specifically within each iteration, different folds of the data are also kept out for validation, while the remaining k-1 folds are retained for learning. k samples of the performance metric will then be available for each models to be evaluated. Aggregation measures such as averaging can be further performed to highlight model performance comparison, otherwise the samples can be used to support a statistical hypothesis test. [37]. Table 12 shows the comparative result between Naive Bayes, J48 and the IBk classifier. From Table 12, the J48 Classifier is shown to have the highest accuracy percentage of 99.27%, and the most outstanding agreement of Kappa Statistic at 0.98 using the 10-fold Cross-Validation method.

TABLE XII
COMPARATIVE RESULT BETWEEN CLASSIFIERS USING 10-FOLD CROSS-VALIDATION

Classifier	Classification Accuracy (%)	Kappa Statistic
Naïve Bayes	97.27	0.92
J48	99.27	0.98
IBk	96.80	0.90

2) Statistical Tests: A statistical test was performed in this study to identify whether two machine learning models are statistically significantly different or whether one of them is better than another. Specifically, the paired corrected t-test was performed to the 119,750 data. Naive Bayes, J48 and IBk classifier are evaluated against the eKasih dataset with a twin-tailed confidence of 0.05 (95%). In this experiment, Naive Bayes classifier serves as the baseline using accuracy (percent correct) as the basis of comparison.

From Table 13, the Naive Bayes is the base for comparison marked as (1) has the accuracy of 97.27% in relation to the problem. This result is compared to the J48 classifier which is marked as (2) and the IBk classifier, marked as (3). The asterisk character (*) next to IBk results indicate that the results are significantly different from the Naive Bayes results. A lower case '*v*' next to J48 indicate that the results are significantly better from Naive Bayes results with 99.27% accuracy. This shows that J48 classifier is the best performer and the result is statistically significant at the 0.05 level.

TABLE XIII
CROSS-VALIDATION T-TEST RESULTS

Tester: Paired Corrected T-Test			
Analysing: Percent_correct			
Dataset: 1			
Resultsets: 3			
Confidence: 0.05 (two tailed)			
Date: 5/26/2018			
Dataset	**(1) Naïve Bayes**	**(2) J48**	**(3) IBk**
eKasih	97.27	99.27 v	96.80 *
	(v/ /*)	(1/0/0)	(0/0/1)

IV. CONCLUSION

This study provides a comparison of performance between three classification methods: Naïve Bayes, Decision Tree (J48) and k-Nearest Neighbor (IBk) in classifying B40 households using eKasih dataset. The classification accuracy of these three methods are compared to each other. Prior to performance comparison, several pre-processing techniques such as data cleaning, feature engineering, feature selection, sampling, and parameter tuning were first conducted. After obtaining optimal values of each classifier, a series of experiments were carried out using 10-fold cross validation. Statistical relevance of the experimental results is determined by the paired t-test based on ten-fold cross-validation and the results demonstrate the Decision Tree model is statistically significant and outperformed other classifiers. The eKasih dataset consisted of missing values and outliers. This corresponds well with Decision Tree model which are less sensitive to missing values and outliers since splitting of data for tree building is based on proportion of samples within the split ranges and not on absolute values. In addition, Decision Tree performs pruning after tree generation, resulting in reduction of tree structure complexity as well as reducing chances of overfitting, thus producing higher accuracy on the B40 Classification. Therefore, we can conclude that B40 Classification using eKasih dataset will perform better if Decision Tree (J48) is used instead of Naive Bayes and k-Nearest Neighbor.

ACKNOWLEDGMENT

The authors would like to thank UKM for providing the funding for the project under the grand challenge LAB40 research grant of DCP-2017-015/1 and special thanks to Society Wellbeing Department, Implementation Coordination Unit of Prime Minister's Department (ICU JPM) for the data set used in the experiment.

REFERENCES

[1] EPU, E. P. U. (2013). Tenth Malaysia Plan. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

[2] Selvaratnam, D. P., Tin, P. B., Bakar, N. A., Idris, N. A. H., & Berma, M. (2017). Social capital accumulation in Malaysia. *e-Bangi*, *3*(1).

[3] Roshaniza, N. A. B. M., & Selvaratnam, D. P. (2015). Gross Domestic Product (GDP) Relationship with Human Development Index (HDI) and Poverty Rate in Malaysia. *Prosiding Perkem*, *10*, 211-217.

[4] Ali, A. F. M., Rashid, Z. A., Johari, F., & Aziz, M. R. A. (2015). The effectiveness of Zakat in reducing poverty incident: An analysis in Kelantan, Malaysia. *Asian Social Science*, *11*(21), 355.

[5] Economic Planning Unit. (2015). Eleventh Malaysia Plan : Anchoring Growth on People. Rancangan Malaysia Kesebelas (Eleventh Malaysia Plan) : 2016-2020.

[6] DOSM. (2017). Department of Statistics Malaysia Press Release Report of Household Income and Basic Amenities Survey 2016, (October). Retrieved from https://www.dosm.gov.my/v1/index.php?r=column/pdfPrev&id=RUZ5REwveU1ra1hGL21JWVlPRmU2Zz09

[7] Holliday, J. D., Sani, N., & Willett, P. (2015). Calculation of substructural analysis weights using a genetic algorithm. *Journal of Chemical Information and Modeling*, *55*(2), 214-221.

[8] Sani, N.S. (2017). The Use of Data Fusion on Multiple Substructural Analysis Based GA Runs. *J. Appl. Environ. Biol. Sci.*, *7*(2S)30-36, 2017

[9] Rahman, A. H. A., Ariffinv, K. A. Z., Sani, N. S., & Zamzuri, H. (2017). Pedestrian Detection using Triple Laser Range Finders. *International Journal of Electrical and Computer Engineering (IJECE)*, *7*(6), 3037-3045.

[10] Pareek, P., & Prema, K. V. (2012). Classifying the population as BPL or non-BPL using Multilayer Neural Network. *International Journal of Scientific and Research Publications*, *2*(1), 2250–3153.

[11] Zakaria, N. H., Hassan, R., Othman, M. R., Zakaria, Z., & Kasim, S. (2017). A Review on Classification of the Urban Poverty Using the Artificial Intelligence Method. *Journal of Asian Scientific Research*, *7*(11), 450.

[12] Thoplan, R. (2014). Random forests for poverty classification. *International Journal of Sciences: Basic and Applied Research (IJSBAR), North America*, *17*.

[13] Redjeki, S., Guntara, M., & Anggoro, P. (2015). Naive Bayes Classifier Algorithm Approach for Mapping Poor Families Potential. *International Journal of Advanced Research in Artificial Intelligence*, *4*(12), 29–33.

[14] Nataša, P. (2016). *Poverty analysis using machine learning methods* (Bachelor Thesis). Comenius University, Bratislava, Slovakia.

[15] Terano, R., Mohamed, Z., & Jusri, J. H. H. (2015). Effectiveness of microcredit program and determinants of income among small business entrepreneurs in Malaysia. *Journal of Global Entrepreneurship Research*, *5*(1), 22.

[16] Siwar, C., Idrus, S., Idris, N. D. M., & Zahari, S. Z. Poverty Mapping and Characterizing the Poor Using Geographical Information System: Case Study in Terengganu, Malaysia. [10] Webb, G. I. (2010). Data Preparation. *Encyclopedia of Machine Learning*, 259–260. https://doi.org/10.1007/978-0-387-30164-8_194

[17] Nawi, N. M., Hussein, A. S., Samsudin, N. A., Hamid, N. A., Yunus, M. A. M., & Ab Aziz, M. F. (2017). The Effect of Pre-Processing Techniques and Optimal Parameters selection on Back Propagation Neural Networks. *International Journal on Advanced Science, Engineering and Information Technology*, 7(3), 770-777.

[18] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[19] SamsiahSani, N., Shlash, I., Hassan, M., Hadi, A., & Aliff, M. (2017). Enhancing Malaysia Rainfall Prediction Using Classification Techniques. *J. Appl. Environ. Biol. Sci*, *7*(2S), 20-29.

[20] Zainudin, S., Jasim, D. S., & Abu Bakar, A. (2016). Comparative analysis of data mining techniques for Malaysian rainfall prediction. *International Journal on Advanced Science, Engineering and Information Technology*, *6*(6), 1148-1153.

[21] Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462.

[22] Kurniawan, R., Nazri, M. Z. A., Irsyad, M., Yendra, R., & Aklima, A. (2015, August). On machine learning technique selection for classification. In *Electrical Engineering and Informatics (ICEEI), 2015 International Conference on* (pp. 540-545). IEEE.

[23] Shreem, S. S., Abdullah, S., & Nazri, M. Z. A. (2016). Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm. *International Journal of Systems Science*, *47*(6), 1312-1329.

[24] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16-28.

[25] Alhutaish, R., & Omar, N. (2017). Feature Selection for Multi-label Document Based on Wrapper Approach through Class Association Rules. *International Journal on Advanced Science, Engineering and Information Technology*, *7*(2), 642-649.

[26] Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2015). Classification with class imbalance problem: A review. *International Journal of Advances in Soft Computing and Its Applications*, *7*(3), 176–204.

[27] Holliday, J., Sani, N., & Willett, P. (2018). Ligand-based virtual screening using a genetic algorithm with data fusion. *Match: Communications in Mathematical and in Computer Chemistry*, *80*, 623-638.

[28] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, *61*, 863-905.

[29] Berend, D., & Kontorovich, A. (2015). A finite sample analysis of the Naive Bayes classifier. *Journal of Machine Learning Research*, *16*, 1519-1545.

[30] Sewaiwar, P., & Verma, K. K. (2015). Comparative study of various decision tree classification algorithm using WEKA. *International Journal of Emerging Research in Management &Technology*, *4*, 2278-9359.

[31] Wager, S., & Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*.

[32] Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130.

[33] Thanh Noi, P., & Kappas, M. (2017). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. Sensors, 18(1), 18.

[34] Hu, L. Y., Huang, M. W., Ke, S. W., & Tsai, C. F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus, 5(1), 1304.

[35] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

[36] Cao, H., Sen, P. K., Peery, A. F., & Dellon, E. S. (2016). Assessing agreement with multiple raters on correlated kappa statistics. *Biometrical Journal*, *58*(4), 935-943.

[37] Refaeilzadeh, P., Tang, L., & Liu, H. (2016). Cross-validation. *Encyclopedia of database systems*, 1-7. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4978658/