Available *online* at : http://jitce.fti.unand.ac.id/
JITCE (Journal of Information Technology and Computer Engineering)
| ISSN (Online) 2599-1663 |

Machine Learning Application for Classification Prediction of Household's Welfare Status

Nofriani

BPS-Statistics of Bengkulu Province, Bengkulu, Indonesia

ARTICLE INFORMATION

Received: June 5th, 2020
Revised: July 9th, 2020
Available online: September 30th, 2020

KEYWORDS

Machine learning, python, scikit-learn, integrated database, random forest algorithm

CORRESPONDENCE

Phone: +62 (0736) 349118
E-mail: nofriani@bps.go.id

A B S T R A C T

Various approaches have been attempted by the Government of Indonesia to eradicate poverty throughout the country, one of which is equitable distribution of social assistance for target households according to their classification of social welfare status. This research aims to re-evaluate the prior evaluation of five well-known machine learning techniques; Naïve Bayes, Random Forest, Support Vector Machines, K-Nearest Neighbor, and C4.5 Algorithm; on how well they predict the classifications of social welfare statuses. Afterwards, the best-performing one is implemented into an executable machine learning application that may predict the user's social welfare status. Other objectives are to analyze the reliability of the chosen algorithm in predicting new data set, and generate a simple classification-prediction application. This research uses Python Programming Language, Scikit-Learn Library, Jupyter Notebook, and PyInstaller to perform all the methodology processes. The results shows that Random Forest Algorithm is the best machine learning technique for predicting household's social welfare status with classification accuracy of 74.20% and the resulted application based on it could correctly predict 60.00% of user's social welfare status out of 40 entries.

INTRODUCTION

According to the 18th World Congress of The International Society of Gynecological Endocrinology [1], Indonesia is one of 145 developing countries in the world. A very common criterion that makes a country categorized as "developing" is the existence of poverty in it [2].

Poverty has also become a vast-spreading issue in Indonesia for over a decade, and it is still one of the most discussed and debated issues to this day. Nonetheless, the Government of Indonesia has attempted various methods to eradicate poverty throughout the country. One of the approaches the Government has attempted is the distribution of equitable social assistance across the archipelago for the target households; those categorized as worth receiving social assistance according to the Integrated Database.

Integrated Database consists of an extensive amount of variables contributing to the classification of a household's social welfare status which thereby concludes whether or not the household is worth receiving social assistance [3]. Bengkulu Province, as one of the less developed provinces in Indonesia, also has its own share of Integrated Database and has so far distributed social assistance across the region based on the classifications of household's social welfare status in Integrated Database.

Machine learning has become one of the most important fields within development organizations that are looking for innovative ways to grasp data assets to help the business attain a new level of understanding [4]. One of the most studied topics of research on supervised machine learning is the case of classifications. For with the explosion of online electronic documents in recent times, it is becoming necessary assistance to people in searching, organizing and collecting related documents [5]. One of the branches of classification study is prediction algorithm analysis. On the other hand, Integrated Database is also a popular case study for researches in Indonesia, shown by a good number of researches on it, and particularly because of its vastly varying contributing variables and a large number of data sets.

Previous works have used the Integrated Database as a case study. A research used Naïve Bayes Algorithm to classify poor household's social welfare status in Integrated Database of 2011 into poor and very poor, using 16 contributing attributes [6]. Another research compared Naïve Bayes Classifier and C4.5 Algorithm in classifying poverty level from Integrated Database of 2011 using 14 contributing attributes [7].

There have also been a high amount of researches on comparisons of the classification algorithms using a large-size data set like Integrated Database. A research compared some widely used machine learning algorithms namely Random Forests Algorithm, Support Vector Machines, Linear Discriminant Analysis and K-Nearest Neighbour on multiple real data examples from mental studies. The research found that better performance in one or a few instances does not necessarily imply so on an average or on a population level and simulation studies may be a better alternative for objectively comparing the performances of machine learning algorithms [8]. The research's finding on the effect of simulation (repetitive runs of algorithm) on the final result of the comparative analysis on classifiers' performances partially inspires a further comparative evaluation of machine learning algorithms performed in the current research.

Most recently, a research used Integrated Database of 2015 as a case study to compare the performances of five supervised machine learning techniques in predicting household's social welfare status, on data set containing 23,872 fields and 45 contributing attributes. Weka Application Version 3.8.2 was used to run all the algorithms and evaluate the performance of each one. The five techniques of supervised machine learning were Naïve Bayes Classifier, Random Forest Algorithm, Support Vector Machines, K-Nearest Neighbor Classification, and C4.5 Algorithm. They were evaluated in terms of their classification accuracies, precision and recall scores, confusion matrices and AUC values. The results concluded that among these five, Random Forest Algorithm gave the best performance with the classification accuracy of 73.42%, precision value of 0.696, recall value of 0.734, and AUC value of 0.850 [9]. This goes in line with the factual theory that the Random Forests Framework has been extremely successful as a general purpose classification and regression method [10]. However, this research is still expandable, for the chosen algorithm (Random Forest Algorithm) can be implemented into a real supervised machine learning application to predict a household's social welfare status.

This research aims to continue and implement the results of the previous research mentioned above. It aims to further evaluate the performances of each of the five algorithms used in the previous research. The re-evaluation is deemed necessary because further observations indicated that in the data set used by the previous research, a quite large number of duplicates were found, which the previous research failed to notice. It is possible that these duplicates unequivocally cause biased results in the performances of the algorithms applied to the prior data set. Therefore these duplicates must be removed and the algorithms must be re-evaluated again afterwards to see if the best performing one still comes from the one with best classification accuracy. Furthermore, this research also aims to implement the best-performing machine learning technique which was re-chosen through re-evaluation processes, into a real executable machine learning application to determine whether the algorithm really gives good predictions on new cases of households' social welfare statuses.

The detailed objectives of the current research are to briefly reevaluate the previously evaluated five supervised machine learning techniques from the previous research, to particularly evaluate the reliability of the best performing one resulted in re-evaluation in predicting new data set and implement it into a real executable application, and to generate a simple portable executable machine learning application on classification predictions.

The scope of the current research only includes the five supervised machine learning techniques from the previous research; i.e. Naïve Bayes Classifier, Random Forest Algorithm, Support Vector Machines, K-Nearest Neighbor Classification, and C4.5 Algorithm. They were chosen because of their popularity and frequent usage in data classification analysis by supervised machine learning. Furthermore, it does not attempt to publicize the reliability and validity of Integrated Database of 2015, hence the resulted executable application shall not be distributed to the public. Moreover, the Integrated Database used in this research is limited to the provincial level, namely Bengkulu, as one of the less developed provinces in Indonesia, wherein poverty is still a major issue.

The previous research used an open source tool, namely Weka Application Version 3.8.2 to run all of the algorithms and evaluate the results of each one. It is an independent platform which is written in Java™ language and contains a graphical user interface to interact with data files and produce visual results [11]. Even though it is an open-source software, and contains a collection of machine learning algorithms for data mining tasks [12]; Weka is a bit slow in performing tasks on large-size data. During the previous research, the data size was only 3.88 MB. But it took 78.61 seconds to build a Support Vector Machines (SVM) classifier model and 23.63 seconds to build a Random Forest Algorithm classifier model. More importantly, Weka does not provide a feature to build an executable application of its algorithms. Therefore, it cannot serve one of the objectives of the current research, which is to develop a software application.

For the above mentioned shortcomings of Weka, the current research chooses a more powerful platform to analyze and implement supervised machine learning techniques, i.e. Python Programming Language. It is one of the most popular programming languages of recent times. Aside from being open-source and high-level, Python also supports object oriented, imperative, functional and procedural development paradigm [13]. Since Python is a general-purpose language, it can do a set of complex machine learning tasks and enable the user to build prototypes quickly that allow them to test their product for machine learning purposes [14]. This surely benefits one the intended goals of this research which is to build a portable application for Python allows users to quickly build prototypes.

Being a very powerful and flexible programming language, Python provides a large number of libraries for machine learning. One of which is Scikit-Learn. Scikit-learn exposes a wide variety of machine learning algorithms, both supervised and unsupervised, using a consistent, task-oriented interface, thus enabling easy comparison of methods for a given application; making it a considerable choice for algorithms evaluation performed in the next steps of this research. Scikit-Learn is known of its being very easy to use, yet implementing many machine learning algorithms efficiently [15]. Another reason for the use of Scikit-Learn is because it is the fastest working Library in Python when it comes to performing machine learning tasks.

Scikit-Learn performs the fastest among other libraries like MLPy, PyBrain, PyMVPA, MDP (Markov Decision Process), and Shogun on four of machine learning tasks performed using six algorithms; Support Vector Classification, Lasso (LARS), Elastic Net, K-Nearest Neighbors, PCA (Principal Component Analysis) and K-Means [16].

METHOD

This paper provides a brief review on the five supervised machine learning techniques used throughout the research in classifying the classes of social welfare status.

C4.5 Algorithm

C4.5 Algorithm is used to generate a decision tree (making decision), based on a certain sample of data (univariate or multivariate predictors). It uses the concept of information entropy, wherein the training data is a set of already classified samples which consist of a dimensional vector that represent attribute values of features of the sample.
C4.5 algorithm is an extension of the earlier IDE3 algorithm, another type of decision tree classifier. However, C4.5 algorithm is proven to outperform the IDE3 algorithm in the area of decision trees [17], due to a number of improvements to IDE3. Such as handling both continuous and discrete attributes, handling training data with missing attribute values, and pruning trees after creation.

KNN Classification

The K-Nearest Neighbors (KNN) Algorithm is a non-parametric method used for classification cases (KNN Classification) and regression cases (KNN Regression). In both cases of KNN Classification and KNN Regression, the input consists of the k closest training examples in the feature space. Specifically, the output in the case of KNN Classification is a class membership. An object is classified by a plurality vote of its neighbors [18], with the object being assigned to the class which is the most common among its k nearest neighbors. In this algorithm, the k is a positive integer, typically small; and if k equals 1, then the object will be assigned to the class of that single nearest neighbor.

Naïve Bayes Classifier

A Naïve Bayes Classifier is a family of simple probabilistic classifier in statistics. It is based on applying Bayes' theorem with strong independence assumptions between the features [19]. All Naïve Bayes Classifiers assume that the value of a particular feature is independent of the value of any other feature, regardless the correlations between each feature. This makes the assumptions used in the Naïve Bayes Classifiers oversimplified, which is why it's called *naive*. They simply make assumptions that may or may not turn out to be correct. Nevertheless, they have worked quite well in many complex real-world situations [20].

Random Forest Algorithm

Random Forest (RF) Algorithm is a classification algorithm consisting of a large number of decisions trees. It uses bagging and feature randomness when building each individual tree to try creating an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree [21]. In general, the more tree there are in the forest, the more robust the prediction will be, and thus the higher the resulted accuracy obtained. Random decision forests also correct the decision trees' habit of overfitting to their training set. However, overfitting in RF Algorithm might happen if there is too much noise in the data. [22]

Support Vector Machines

The Support Vector Machine (SVM) algorithm is a popular machine learning tool that offers solutions for both classification and regression problems. Its training algorithm builds a model that assigns new data set to one category or making it a non-probabilistic binary linear classifier.
An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. [23]

The flow of methodology of the current research is depicted in Figure 1.

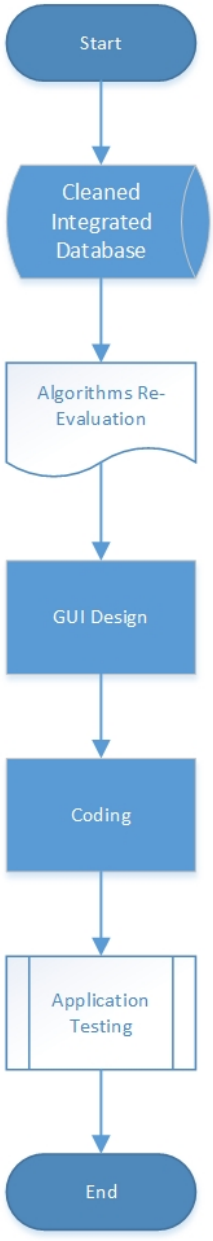

Figure 1. General Flow of Methodology

https://doi.org/10.25077/jitce.4.02.72-82.2020

The work flow includes; acquiring the CSV file of cleaned Integrated Database of 2015 as the data set, removing duplicates from it, briefly re-evaluating the performances of all five algorithms on the predicting the classification labels in the data set, designing the Graphical User Interface for the application based on the re-evaluation result, writing Python codes for the application whilst further evaluating the chosen algorithm's performance, and testing the performance and accuracy of the application using Black Box Testing and Accuracy Measurement.

The current research used secondary data source, i.e. Integrated Database for Bengkulu Province of 2015. The data is in a form of CSV file and had been pre-processed during the previous research. Several attributes were deleted for not all of the attributes are useful for classification [24]. It consists of one attribute as classification label, i.e. Household's Welfare Status; and 45 all-nominal-data attribute labels that contribute to the classification label. The detailed information on the classification and attribute labels used in this research is depicted in Table 1.

Table 1. Classification & Attribute Labels of Integrated Database

Variable	Explanation
Class: Social Welfare Status	1: Household in the lowest 10 percent of welfare status. Included in decile 1 (Very poor) 2: Household in the lowest 11to 20 percent of welfare status. Included in decile 2 (Poor) 3: Household in the lowest 21 to 30 percent of welfare status. Included in decile 3 (Near poor) 4: Household in the lowest 31 to 40 percent of welfare status. In decile 4 (Vulnerable)
X_1	Highest level of education
X_2	Occupation or business field
X_3	Status on primary occupation
X_4	Status of residential building
X_5	Status of residential land
X_6	Residential building's floors area
X_7	Residential building's floors type
X_8	Residential building's walls type
X_9	Residential building's walls condition
X_{10}	Residential building's roof type
X_{11}	Residential building's roof condition
X_{12}	Number of bedroom in residential building
X_{13}	Source of drinking water
X_{14}	Way to access drinking water
X_{15}	Primary lighting source
X_{16}	Type of installed electrical power
X_{17}	Cooking fuel/utensil
X_{18}	Type of defecation facility
X_{19}	Toilet type
X_{20}	Type of final fecal disposal facility
X_{21}	Ownership status of gas cylinders with a capacity of 5.5 kg or above
X_{22}	Ownership status of the refrigerator
X_{23}	Ownership status of the air conditioner
X_{24}	Ownership status of water heater
X_{25}	Ownership status of the house phone
X_{26}	Ownership status of television
X_{27}	Ownership status of computer or laptop
X_{28}	Ownership status of bicycle
X_{29}	Ownership status of the motorcycle
X_{30}	Ownership status of the car
X_{31}	Ownership status of the boat
X_{32}	Ownership of outboard motor
X_{33}	Ownership of motorboat
X_{34}	Ownership of ship
X_{35}	Number of the owned active phone number
X_{36}	Number of owned LCD TV
X_{37}	Ownership status of land asset
X_{38}	The total area of owned land asset
X_{39}	Ownership status of the house beside the residential building
X_{40}	Number of owned cow
X_{41}	Number of owned buffalo
X_{42}	Number of owned horse
X_{43}	Number of owned pig
X_{44}	Number of owned goat
X_{45}	Number of a household member

Source: Agency of Social Affairs of Bengkulu Province

Before proceeding with the algorithm's implementation, the data set from the original database was again observed and cleaned. The observation found that there were duplicates of 3,566 fields out of 23,872 fields of original database, which the previous research failed to notice. This lead to a second consideration to re-evaluate the result's reliability on the algorithms comparison conducted by the previous research, for the existence of duplicates on a data set may lead to biased analysis. One of the biggest challenges in data analytics is to discover and repair dirty data; failure to do this can lead to inaccurate analytics and unpredictable conclusions [25]. Removing duplicate records is a crucial step in data cleaning process [26].

Thereupon, in this research those duplicates were removed from the database, giving remaining 20,306 fields of data set, or as much as 84.88% of the original database. This duplicates removal reduced the amount of fields of each classification labels almost equally, as shown in Table 2. The absolute value of each class frequency is not shown for the confidentiality of Integrated Database of 2015 for Bengkulu Province.

Table 2. The Decrease of Class Frequency of Household's Welfare Status after Duplicates Removal

Class	Decrease of Class Frequency
1	13.45%
2	17.15%
3	22.52%
4	14.44%
Weighted Average	15.12%

The equitable decrease of each classification frequency shown above ensures that the new dataset does not have imbalance classification labels, hence give a relevant evaluation on the next steps of this research.

The step of duplicates removal is performed in order to ensure that the training process is more reliable and the resulted conclusions are more accurate. As a result, those five algorithms from the previous research should be re-evaluated due to a

hypothesis that the classification accuracy of each may be higher or lower if the duplicates in the original database were removed. It is also to be more certain on the results of the previous research by ensuring that a new method (different platform) gives similar results.

The current research then re-train the Integrated Database using all five supervised machine learning techniques on Python code lines. All the processes of data training, testing, and prediction were performed using the help of The Jupyter Notebook, an open-source web application that allows the users to create and share documents that contain live code, equations, visualizations and narrative text. The uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, etc [27].

After all the algorithms were re-trained and re-tested, the resulted models were compared; then the one with the highest accuracy was chosen to be used as the basis of classifier model for the intended machine learning application. The application is named "WeCa", short for Welfare Status Classification.

Before proceeding to writing the code, the GUI (Graphical User Interface) for the application was first designed. Figure 2 shows the design of GUI for WeCa application.

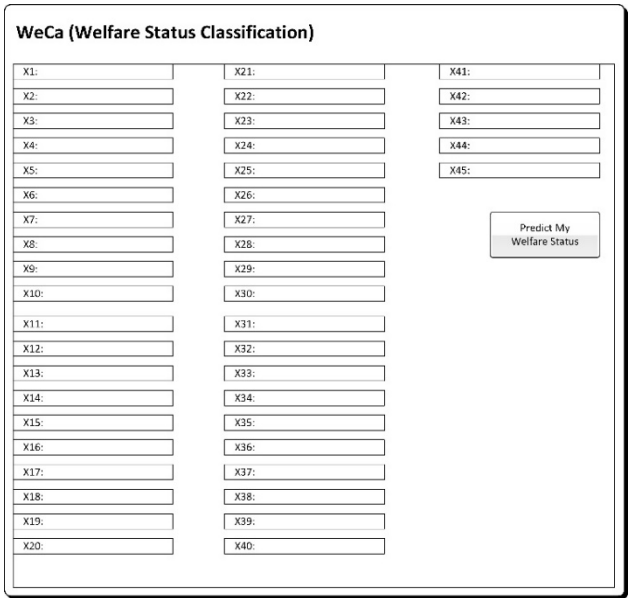

Figure 2. General GUI Design for WeCa Application

The development of this application was performed in Python code lines on Jupyter Notebook. However, Jupyter Notebook uses IPYNB as the extension of the source code's document. Therefore, the resulted file was then converted to PY file as the original file extension for Python Projects. It was to enable the compilation of the source code's document into an executable software application (EXE). The converting process to PY file was done using the feature provided in Jupyter Notebook. While the compiling it to EXE (executable) file was done using PyInstaller library provided by Python Programming Language.

RESULTS AND DISCUSSION

Accuracy Comparison between Weka and Python

The classification accuracies of all five classifier models obtained using Weka application and Python code lines are compared. Table 2 presents the classification accuracy of each model obtained using Weka application and Python code lines.

Table 2. Classification Accuracy of Weka and Python

Machine Learning Technique	Classification Accuracy with Original Database	
	On Weka	On Python
C4.5 Algorithm	72.07%	63.65%
KNN Classification	59.32%	64.66%
Naïve Bayes Classifier	62.34%	62.56%
Random Forest Algorithm	73.42%	72.19%
Support Vector Machines	65.40%	66.37%

Source: Classification Accuracies from Previous and Current Research

The data in Table 2 shows there are differences of classification accuracies obtained using Weka application and Python code lines. Two of the algorithms actually perform less well in Python while the other three perform better; particularly KNN Classification's accuracy which jumps from 59.32% to 64.55%. Nevertheless, generally the classification accuracies become higher on Python's Scikit-Learn Library than the ones obtained on Weka from the previous research. This denotes that in addition to its fast-working response, it is safe to say that Python is a better choice when it comes to supervised machine learning tasks, especially for the five algorithms used in the current research. This goes in line with the fact that Python is evidently more popular than Weka as a machine learning tool [28].

Algorithms Comparison

After comparing their performances on Weka and Python, all five classifier models were also briefly re-evaluated on the latter based on their respective classification accuracy. It is to support the validity of classification accuracy due to the possible effect of duplicates removal from the original database. Prior to a comparison of different classification algorithms, it is important to consider which comparative measure should be used [29]. Here classification accuracy was chosen as the comparative measure on the five supervised machine learning techniques, as it is indeed the most commonly used to compare classifiers' performances.
Table 3 presents the classification accuracy of each model using Python code lines on the original data set and the one without duplicates. Figure 3 depicts the data visualization of Table 2. Figure 4 depicts the data visualization of Table 3.

Table 3. Classification Accuracy of Each Supervised Machine Learning Techniques Using Python

Machine Learning Technique	Classification Accuracy	
	With Original Database	Duplicates Removed
C4.5 Algorithm	63.65%	64.16%
KNN Classification	64.66%	62.92%
Naïve Bayes Classifier	62.56%	63.56%
Random Forest Algorithm	72.19%	71.19%
Support Vector Machines	66.37%	65.58%

Source: Classification Accuracies from Current Research

https://doi.org/10.25077/jitce.4.02.72-82.2020

Available *online* at : http://jitce.fti.unand.ac.id/
JITCE (Journal of Information Technology and Computer Engineering)
| ISSN (Online) 2599-1663 |

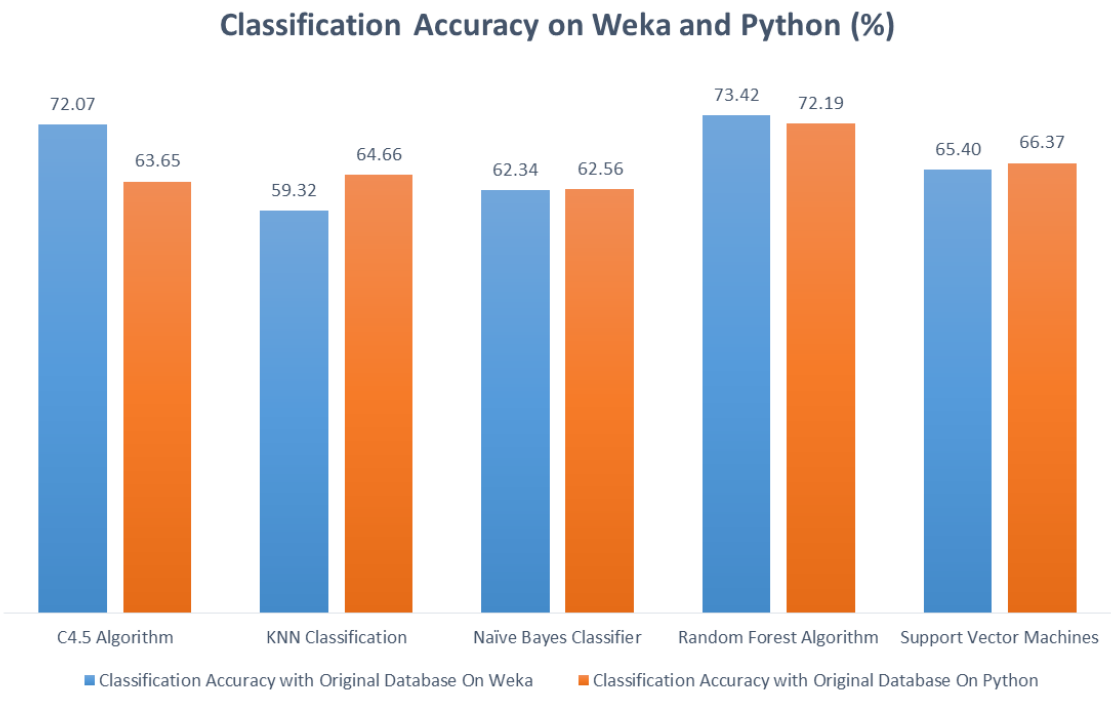

Figure 3. Data Visualization of Classification Accuracy on Weka and Python

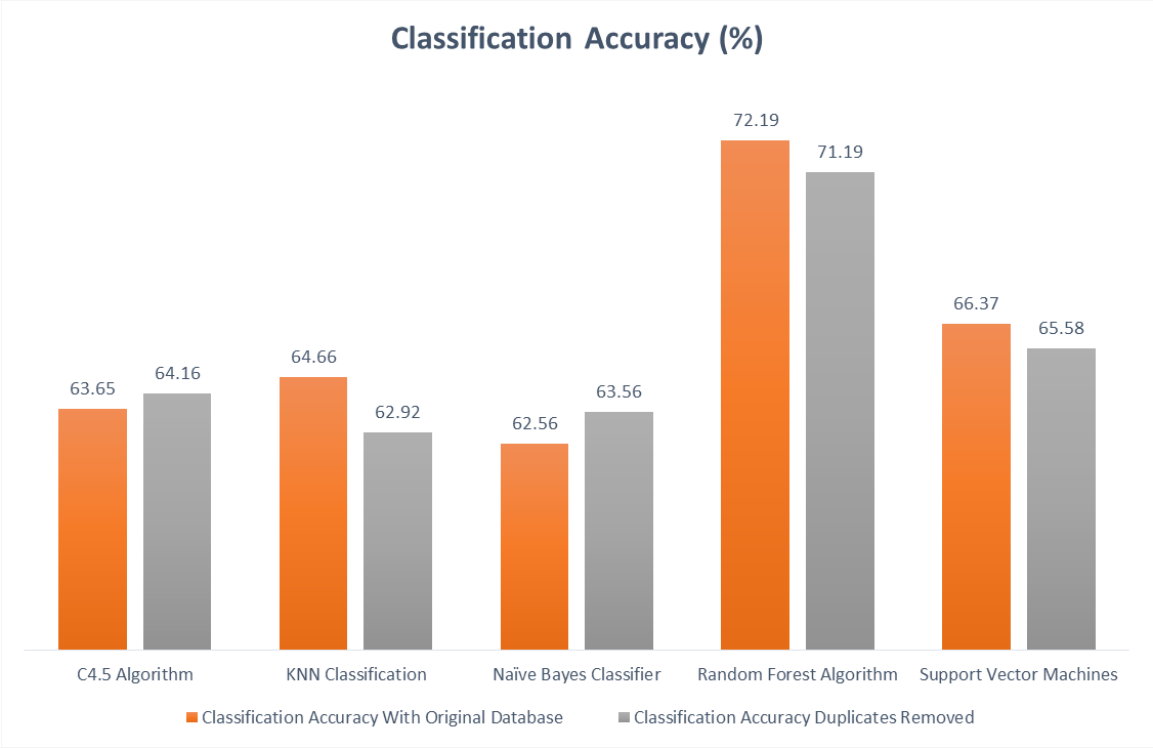

Figure 4. Data Visualization of Classification Accuracy on Python

The data in Table 3 shows that the removal of data duplication in the data set indeed affects the classification accuracy of each algorithm, indicated by different classification accuracy on each classifier model. C4.5 Algorithm and Naïve Bayes Classifier perform better on the data set without duplicates; indicated by their higher respective classification accuracy. While the other three; KNN Classification, Support Vector Machines and Random Forest Algorithm—as the seemingly best performing

one—perform less well; indicated by their lower respective classification accuracy.

The higher classification accuracy is probably because of the duplicates removal after all, for data cleaning can help achieve better results in classification problems [30]. Particularly on Naïve Bayes Classifier, the high rates of duplicates can be quite harmful for its classification accuracy [31]. On the other hand, the lower classification accuracy may be caused by several possible reasons. First, the different split of testing and training set (different fold of *k*-fold cross validation) each time a classifier model runs on the data set may cause different classification accuracy, in spite of removed duplicates. Second, the classification variance of each classifier model may cause that re-running process on different training and testing to yield classifications that differ quite significantly. For theoretically, smaller data set will give higher variance [32], which is what possibly happens in this scenario; removing duplicates also reducing the sample size, thus increasing the variance of classification accuracy.

However, the data shows that Random Forest Algorithm still gives the highest classification accuracy. Either on the original database or the one without duplicates, Random Forest Algorithm is the only algorithm with classification accuracy above 70 percent. Based on this observation, it is concluded that using a different platform, and with or without duplicates, Random Forest Algorithm is still highly likely the best practical machine learning technique in predicting the classification of household's social welfare status in Integrated Database of 2015 for Bengkulu Province. Therefore, it is safe to choose this algorithm as the basis of classifier model for the intended classification.

Resulted Software Application

Figure 5 shows the initial execution of resulted intended application using the chosen algorithm (Random Forest Algorithm) as the basis of classifier model. Figure 6 shows the application's prediction on user's social welfare status classification after the users types on their data and clicks the button to calculate prediction.

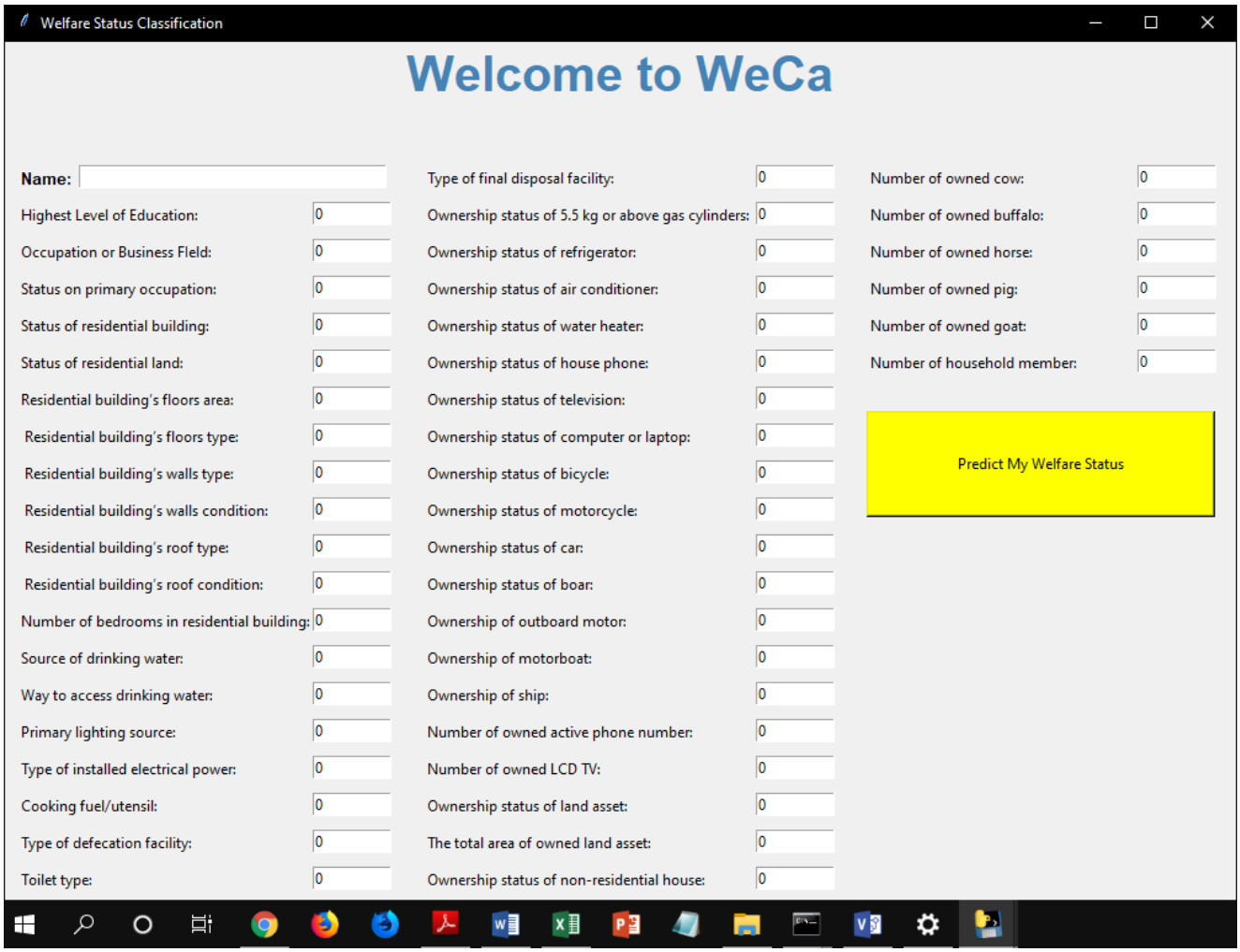

Figure 5. WeCa Application's Initial Run

 https://doi.org/10.25077/jitce.4.02.72-82.2020

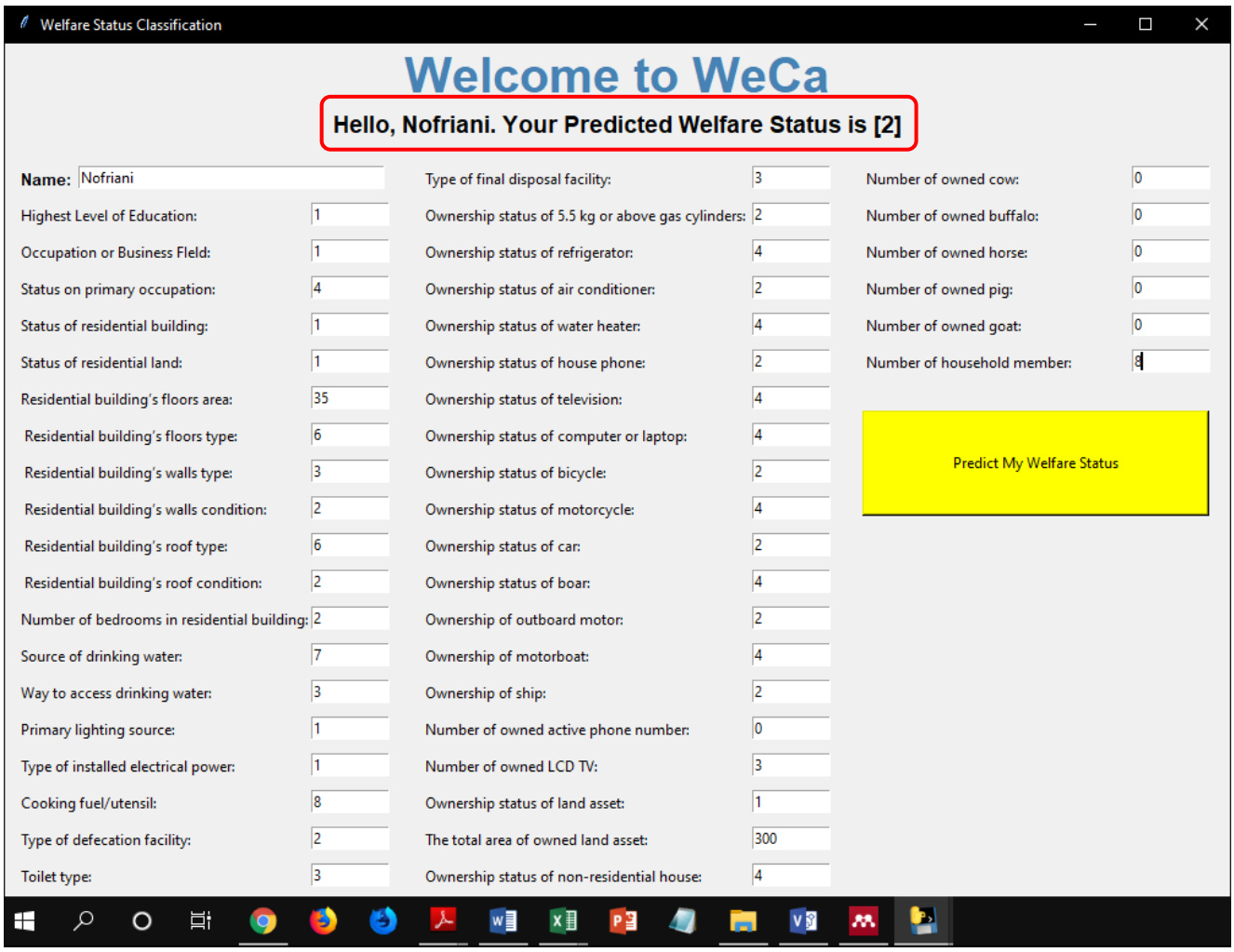

Figure 6. WeCa Application's Prediction

To evaluate whether the resulted application works as expected, this research used Black Box Testing method. It is a software testing method to determine whether the functions in the application work as designed [33]. In the testing process, user gives inputs into the application's functions to observe its response, i.e. output [22]. The testing was performed on 28 November 2019, at 10.10 am. The result is listed in Table 4.

Table 4. Questionnaire of Black Box Testing on WeCa Application

Test	Case	Yes	No
1	If user double clicks on the file, the application will run.	√	-
2	If user types on their name, the text field will show corresponding letters.	√	-
3	If user types on their data on attribute labels, each text field will show corresponding letters.	√	-
4	If user clicks the cross sign ("X") on upper right corner, the application will close.	√	-
5	If user clicks "Predict My Welfare Status" button, the application will show its prediction.	√	-

The result of Black Box Testing indicates that the resulted application is able to properly give corresponding responses according to the user's inputs. Namely 1) when the user double clicks on the application file (EXE), the application opens immediately; 2) when the user types on their name, the application's text field shows corresponding text the user types in; 3) when the user types on their data of social welfare condition, each text field in the application shows corresponding letters (nominal values); 4) when the user clicks the close button ("X") on the upper right corner of the application, the application closes immediately; and 5) when the user clicks the button that reads "Predict My Welfare Status", the application immediately shows its prediction on the user's classification of social welfare status. These behaviors lead to conclusion that the application functions well according to the user's inputs.

Application's Predictions

Aside from Black Box Testing, the application is also tested on how well it can predict the user's social welfare status and how close its accuracy is to the applied algorithm's accuracy during the training and testing process (Random Forest Algorithm). For this purpose, this research used 40 fields of pre-separated real data from Integrated Database of 2015 for Bengkulu Province and tested the prediction accuracy on the application.

The testing is conducted 40 times, consisting 10 times of each classification label (four categories of social welfare status). However, in the series of first trials, the application unexpectedly only predicted "1" ("very poor" category) as the classification label of all 40 test cases, despite the classification accuracy of the classifier model being 71.19%.

Model's Overfitting

A further evaluation was performed to analyze the possible cause of the problem above. It was then found that there was likely an overfitting in the model generated by Random Forest Algorithm. Overfitting refers to a model that models the training data "too well". It happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data [34]. Overfitting is one of the most fundamental problems in machine learning, which in this case is Random Forest Algorithm.

Even though Random Forest Algorithm is known to be able to handle a very large number of input variables without overfitting [35], it has low bias but extremely high variance that does not vanish as the sample size increases, and thus is destined to overfit [36]. If there is high variance, the model is too general and also learns the noise [37]. It causes the model to perform very well with the data used to create it but not necessarily perform equally well with new data [38]. In other words, the model knows the training set very well but cannot be applied to new problems. Therefore, when new data is applied to the model, there's very high likeliness that the model predicts wrong. This is likely the cause of WeCa Application's wrong predictions.

Therefore, an additional procedure is required to avoid the trees in the random forest to overfit [39]. The procedure to handle the overfitting would be through optimizing tuning parameters, i.e.:

1. *n_estimators*, which represents the number of trees in the forest. Usually the higher the number of trees, the better to learn the data and the less likely the model is to be overfitting.
2. *max_features*, which represent the number of features to consider when looking for the best split. The smaller the number, the less likely to overfit. Hence, scaling the features may improve the classification accuracy [40].
3. *random_state*, which represent the random number generator. It controls the random selections of features and samples.

After repeatedly experimenting with the tuning parameters, the classification accuracy became a little higher, i.e. 74.20% and the application gave quite better predictions. Table 5 shows the result of the application's final predictions and the actual classifications of households' social welfare statuses after tuning parameters optimization process. Figure 7 depicts the data visualization of Table 5.

Table 5. Accuracy Testing on WeCa Application's Predictions

Trial	Actual Classification	Application's Classification	Accuracy
1	1	4	0
2		1	1
3		1	1
4		1	1
5		1	1
6		1	1
7		1	1
8		2	0
9		4	0
10		2	0
11	2	2	1
12		1	0
13		1	0
14		2	1
15		2	1
16		4	0
17		2	1
18		2	1
19		2	1
20		2	1
21	3	3	1
22		3	1
23		3	1
24		4	0
25		1	0
26		3	1
27		4	0
28		3	1
29		2	0
30		3	1
31	4	4	1
32		4	1
33		1	0
34		2	0
35		2	0
36		3	0
37		2	0
38		4	1
39		4	1
40		4	1

Source: Application's Classification Accuracies

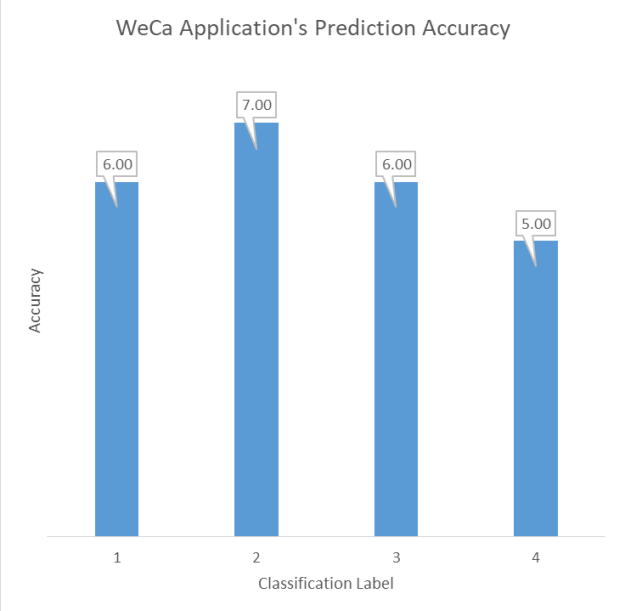

Figure 7. Frequency Plot of WeCa Application's Prediction

Out of 40 entries, the application gave a total of 24 correct predictions, making it 60.00% accurate. Even though it's still below the Random Forest Algorithm's initial accuracy of 74.20%, it is safe to say that the application succeeded in implementing it. Therefore this algorithm is suitable to use in developing a supervised machine learning application in

https://doi.org/10.25077/jitce.4.02.72-82.2020

predicting households' social welfare statuses based on Integrated Database of 2015 for Bengkulu Province.

CONCLUSIONS

Five well-known supervised machine learning techniques have been re-evaluated based on their classification accuracy in predicting household's social welfare status in Integrated Database of 2015 for Bengkulu Province. The data used to generate the classifier models consists of 20,306 fields, with duplicates removed from the original database. The data training and testing were performed in Python code lines using Scikit-Learn library on Jupyter Notebook.

The current research concluded that Random Forest Algorithm gives the best performance in predicting the household's social welfare status with a final classification accuracy of 74.20%. Therefore, this research used the algorithm as the basis for the development of executable Python application to predict a household's social welfare status. According to Black Box Testing and Accuracy Testing, it is concluded that the application was proved to function really well and predict quite well. The application's prediction accuracy score is 60.00%, which is close enough to the classification accuracy of the applied algorithm. But it is still below the algorithm's initial accuracy. Further research is required to analyze other possible causes of this shortcoming. Future work will also be to build a system that provides a feature of relevant feedback in addition to supervised machine learning, and gives prettier GUI.

ACKNOWLEDGMENT

I would love to express my gratitude to Chief Statistician of Bengkulu Province, Dyah Anugrah Kuswardani, M.A. for kindly granting the permission to explore Integrated Database of 2015 for Bengkulu Province. I also thank Dian Putra Nugraha, S.ST., for sharing his pearls of wisdom throughout this research and his companionship. I also thank Dr. Said Mirza Pahlevi, Dr. Muchammad Romzi and Moh. Fatichuddin, M.Eng. for inspiring me to never stop learning and adding something to the world of Data Science. Also to Fatmasari Damayanti, S.Si., M.Si., Novrian Pratama, S.ST., M.Si., and Yosep Oktavianus Sitohang, M.Stat. for their valuable knowledge sharing during the end of this research.

REFERENCES

[1] ISGE. (2018). List of Developing Countries. Retrieved from https://isge2018.isgesociety.com/registration/list-of-developing-countries/.

[2] Ansuategi, A., Greno, P., Houlden, V., Markandya, A., Onofri, L., Picot, H., … Walmsley, N. (2015). The Impact of Climate Change of the Achievement of the Post-2015 Sustainable Development Goals.

[3] TNP2K. (2019). Tentang Data Terpadu PPFM. Retrieved December 3, 2019, from http://bdt.tnp2k.go.id/tentang

[4] Oyedeji, A. O., Salami, A. M., Folorunsho, O., & Abolade, O. R. (2020). Analysis and Prediction of Student Academic Performance Using Machine Learning. *JITCE (Journal of Information Technology and Computer Engineering)*, 4(1), 10–15.

[5] Patra, B. G., Kundu, A., Das, D., & Bandyopadhyay, S. (2012). Classification of Interviews – A Case Study on Cancer Patients. *Proceedings of the 2nd Workshop on Sentiment Analysis Where AI Meets Psychology*, 27–36.

[6] Karyadiputra, E. (2016). Analisis Algoritma Naive Bayes untuk Klasifikasi Status Kesejahteraan Rumah Tangga Keluarga Binaan Sosial. *Jurnal Ilmiah Fakultas Teknik Technologia,* 7(4), 199–208.

[7] Iskandar, D., & Suprapto, Y. K. (2013). Perbandingan Akurasi Klasifikasi Tingkat Kemiskinan antara Algoritma C4.5 and Naive Bayes Classifier. *JAVA Journal of Electrical and Electronics Engineering*, 11(1), 14–17.

[8] Khondoker, M., Dobson, R., Skirrow, C., Simmons, A., & Stahl, D. (2013). A comparison of machine learning methods for classification using simulation with multiple real data examples from mental health studies. *Statistical Methods in Medical Research*.

[9] Nofriani. (2019). Comparations of Supervised Machine Learning Techniques in Predicting the Classification of the Household's Welfare Status. *Pekommas Journal,* 4(1), 43–52. https://doi.org/10.30818/jpkm.2019.2040105.

[10] Denil, M., Matheson, D., & Freitas, N. de. (n.d.). Narrowing the Gap: Random Forests in Theory and In Practice. *Proceedings of the 31 St International Conference on Machine. Learning, Beijing, China, 32*.

[11] Kawelah, W. A. A. S., & Abdala, A. S. E. (2019). A Comparative Study for Machine Learning Tools Using WEKA and Rapid Miner with Classifier Algorithms Random Tree and Random Forest for Network Intrusion Detection. *International Journal of Innovative Science and Research Technology*, 4(4), 749–752.

[12] Waikato, U. (2019). Weka - Machine Learning Software in Java. Retrieved November 25, 2019, from website: http://www.cs.waikato.ac.nz/ml/weka/

[13] Chand, M. (2019). Best Programming Language for Machine Learning. Retrieved November 26, 2019, from https://www.c-sharpcorner.com/article/best-programming-language-for-machine-learning/

[14] Beklemysheva, A. (2019). *Why Use Python for AI and Machine Learning*. https://steelkiwi.com/blog/python-for-ai-and-machine-learning/

[15] Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow* (1st ed.; N. Tachhe, Ed.). United States of America: O'Reilly Media, Inc.

[16] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Bertrand Thirion. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[17] Chauhan, H., & Chauhan, A. (2014). Evaluating Performance of Decision Tree Algorithms 1. *International Journal of Scientific and Research Publication,* 4(4), 1-2.

[18] Cover, T. M., & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions of Information Theory*, 1T-13(1), 21–27.

[19] Murphy, K. P. (2006). Naïve Bayes Classifiers. *University of British Columbia, 18,* 60.

[20] Zhang, H. (2004). The Optimality of Naïve Bayes. *American Association for Artificial Intelligence.*

[21] Yiu, T. (2019). Understanding Random Forest. Retrieved from https://towardsdatascience.com/understanding-random-forest-58381e0602d2
[22] Nofriani. (2013). Pembangunan Mesin Pencari Statistik Berbasiskan Supervised Learning dan Relevant Feedback. Sekolah Tinggi Ilmu Statistik (Polstat STIS).
[23] Wikipedia. (2020). Support Vector Machine. Retrieved July 7, 2020, from https://en.wikipedia.org/wiki/Support_vector_machine
[24] Umarani, V., & Rathika, C. (2019). Predicting Safety Information of Drugs Using Data Mining Technique. *International Journal of Computer Engineering & Technology (IJCET), 10*(2), 89–90.
[25] H., Jesmeen. M. Z., Hossen, J., Sayeed, S., Ho, C. K., K., T., Armanur, R., & Arif, E. M. H. (2018). A Survey on Cleaning Dirty Data Using Machine Learning Paradigm for Big Data Analytics. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(3), 1234–1243.
[26] Selvi, P. (2017). An Analysis on Removal of Duplicate Records using Different Types of Data Mining Techniques: A Survey. *International Journal of Computer Science and Mobile Computing*, 6(11), 38–42.
[27] Jupyter, P. (2019). Project Jupyter. Retrieved November 26, 2019, from https://jupyter.org/
[28] Jović, A., Brkić, K., & Bogunović, N. (2014). *An overview of free software tools for general data mining*.
[29] Pretorius, A., Bierman, S., & Steel, S. J. (2016). A Meta-Analysis of Research in Random Forests for Classification. 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics: International Conference (PRASA-RobMech).
[30] Jeatrakul, P., Wong, K. W., & Fung, C. C. (2010). Data Cleaning for Classification Using Misclassification Analysis. *Journal of Advanced Computational Intelligence and Intelligence Informatics*, 14(3), 297–302.
[31] Kołcz, A., Chowdhury, A., & Alspector, J. (2003). Data Duplication: An Imbalance Problem? *Workshop on Learning from Imbalanced Datasets II, ICML*.
[32] Valencia-Zapata, G., Mejia, D., Klimeck, G., Zentner, M. G., & Ersoy, O. (2017). *A Statistical Approach to Increase Classification Accuracy in Supervised Learning Algorithms.*
[33] Patton, R. (2001). *Software Testing*. United States of America: Sams Publishing.
[34] Brownlee, J. (2019). *Overfitting and Underfitting with Machine Learning Algorithms.* https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/
[35] Biau, G. (2012). Analysis of a Random Forests Model. *Journal of Machine Learning Research*, 13, 1063–1095.
[36] Tang, C., Garreau, D., & Luxburg, U. von. (2018). When do random forests fail? *32nd Conference on Neural Information Processing Systems (NeurIPS).*
[37] Alpaydin, E. (2010). *Introduction to Machine Learning* (2nd ed.; T. Dietterich, Ed.). London: Massachusets Insitute of Technology.
[38] Roßbach, P. (2018). Neural Networks vs. Random Forests – Does It Always Have to be Deep Learning? Germany: Frankfurt School of Finance and Management.
[39] Hastuti, K. (2012). Analisis Komparasi Algoritma Klasifikasi Data Mining untuk Prediksi Mahasiswa Non Aktif. *Seminar Nasional Teknologi Informasi Dan Komunikasi Terapan*, 241–249.
[40] Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques.* Germany: Springer-Verlag Berlin Heildelberg.