JURNAL VOKASI KETEKNIKAN
Vol.2, No. 2, September 2024
e-ISSN: 2986-9226
https://ojs.unm.ac.id/vokasiketeknikan

MEASURING THE PERFORMANCE OF THE NAÏVE BAYES ALGORITHM IN DETERMINING STUDENTS' ABILITY IN THE ITSJava APPLICATION

MANDA ROHANDI[1*], MUHAMMAD YAHYA[2], ABD. MUIS MAPPALOTTENG[3]

Author Affiliations;
[1] Faculty of Engineering, State University of Gorontalo
[2,3] Faculty of Engineering, State University of Makassar
Author Emails;
[1]manda.rohandi@ung.ac.id
[2]Yahya.ftunm@gmail.com
[3]abdulmuism@gmail.com

Abstract. The ITSJava application is used online to increase students' understanding and interest in PBO learning. One of the components of the ITSJava application is the domain/expert model. The expert domain model in the ITS Java application determines whether students can continue to the following material or repeat the material being studied. This research aims to determine the accuracy, precision and sensitivity of the results provided by the Naïve Bayes algorithm in determining the level of students' ability to understand the material in the ITSJava application. The research method uses the Knowledge Discovery in Database (KDD) design, which consists of 1) Data selection, 2) Preprocessing, 3) Transformation, 4) Datamining, and 5) Interpretation/evaluation. This research obtained an accuracy level of 89%, precision = 0.92 and recall = 0.73 if using cross-validation with fold = 10, and an accuracy level of 75%, precision = 1, and recall = 0.37 if using percentage split = 80. These results show that the Naïve Bayes method applied to the ITS Java application can determine the level of students' ability to understand the material.
Keywords: Naïve bayes, accuracy, precission, recall, ITS

A. INTRODUCTION

Expert models in intelligent tutoring systems play an essential role in modern education. This model allows for customization of learning based on individual student needs, which traditional approaches cannot achieve. Providing timely and appropriate support supports more effective and efficient learning. Baker and Siemens (2018) state that using adaptive technology in education can significantly improve student learning outcomes. Expert models in ITS are needed for more focused and in-depth learning. This model uses artificial intelligence algorithms to analyze student data and provide specific, targeted feedback. As technology advances, the capabilities of these systems continue to improve, allowing for more complex and accurate analysis. Nguyen et al. (2020) noted that AI technology in education provides excellent opportunities for innovation in teaching methods. This research highlights the importance of innovation in meeting growing educational needs. Implementing expert models in ITS involves various techniques, including machine learning and natural language processing. These techniques allow the system to understand and adjust teaching methods according to student progress. In addition, this system can identify student weaknesses and strengths more effectively than traditional methods. A study by Woolf (2019) revealed that integrating AI technology into education can increase student engagement and motivation. This shows that advanced technology can produce a more enjoyable and productive learning experience. The presence of expert models in ITS also helps reduce teacher workload by providing consistent automated support. This allows teachers to focus on the more creative and interactive aspects of teaching. Thus, students benefit from both approaches: inspiring human teaching and targeted technological support. According to Zhou et al. (2021), collaboration between AI technology and human teaching can create a more dynamic and effective learning environment. This shows the great potential of synergy between technology and education.

The ITS Java application is an ITS application used in PBO learning. This application is used online to increase students' understanding and interest in PBO learning. This application integrates collaborative learning and learning according to student learning styles into ITS. A critical part of the ITSJava application is the domain/expert model. Domain/expert models in ITS can use various algorithms to evaluate student learning outcomes. These algorithms play a crucial role in identifying students' strengths and weaknesses and providing timely and personalized feedback. Some of the main algorithms used include Decision Trees by Wang et al. (2021), who found that Decision Trees help ITS provide optimal learning routes for students based on their performance and preferences. This algorithm allows the system to provide clear and structured guidance for achieving learning goals. Machine Learning techniques algorithms such as Support Vector Machines (SVM) by Khosravi et al. (2022) show that using machine learning techniques can significantly improve the accuracy of predictions and personalization in ITS. This allows the system to adapt the teaching approach to each student's unique needs. Machine learning algorithms enable complex data analysis and produce more accurate assessments of student performance. According to Anderson et al. (2020), using advanced algorithms in ITS can improve the quality and accuracy of evaluating student learning outcomes.

Apart from the algorithm above, Naïve Bayes is an algorithm that is also widely applied in the ITS expert domain. Naïve Bayes is an algorithm applied in the expert domain of the ITS Java application to determine whether students can

JURNAL VOKASI KETEKNIKAN
Vol.2, No. 2, September 2024
e-ISSN: 2986-9226
https://ojs.unm.ac.id/vokasiketeknikan

continue to the following material or repeat the material being studied. This algorithm was chosen because Bayesian Networks is one of the widely used algorithms in ITS expert systems. Naïve Bayes is a simple classification and statistical classifier method that can predict the probability of belonging to a class. Naïve Bayes, based on Bayesian theory, predicts that the value of an attribute is independent of the value of other attributes (Pebdika et al., 2023). This algorithm models the probability of relationships between various factors that influence student learning outcomes. Using Bayesian Networks, ITS can better predict future student performance based on historical data. A study by Mislevy et al. (2019) shows that Bayesian Networks are very effective in analyzing educational data and providing deep insights into student learning processes. This shows the ability of this algorithm to help teachers and students understand the dynamics of learning better. The reliability of the Naïve Bayes algorithm in determining the level of students' ability to understand the material needs to be done. This is because the reliability of Naïve Bayes describes the validity, level of performance, and trustworthiness of the results provided. Therefore, this research aims to determine the accuracy, precision and sensitivity of the results provided by the Naïve Bayes algorithm in determining the level of students' ability to understand the material in the ITS Java application.

B.METHOD

This research uses the WEKA application to process and analyze the accuracy of the Naïve Bayes algorithm on student learning outcomes data. This research stage refers to the design of Knowledge Discovery in Database (KDD) (Pebdika et al., 2023). This stage consists of:

1. Data selection
   Selection data comes from a set of operational data that must be carried out before the data extraction stage in KDD. Based on operational data, the results of selecting data to be used in the data mining stage are displayed on one page.
2. Preprocessing
   The data cleaning method, which is the focus of KDD, is carried out before proceeding to the next stage. This stage includes deleting duplicate data, checking inconsistent data, and correcting errors in data, such as printing errors.
3. Transformation
   Coding is a data transformation that is selected to be suitable for data mining steps. KDD coding is a creative step that depends on the data type or model taken from the data set.
4. Datamining
   This step uses techniques or methods to find interesting patterns or results. Data mining techniques, methods, and algorithms vary, and determining the suitable method or algorithm depends on the objectives and all stages in KDD.

JURNAL VOKASI KETEKNIKAN
Vol.2, No. 2, September 2024
e-ISSN: 2986-9226
https://ojs.unm.ac.id/vokasiketeknikan

5. Interpretation/Evaluation
   The data model obtained from the data mining step can provide results that are easily understood by those interested. This section aims to determine whether the policy or information obtained contradicts the information.

C.RESULT AND DISCUSSION

1. Data selection

Selection data is taken from all attributes of the original data contained in the database. The data originating from the database is then taken and saved in .arff format, which will then be selected according to what is required. The following is an image of the results of selecting student learning outcomes data in .arff format.

Figure 1. Selected learning outcomes dataset

2. Preprocessing

This stage includes removing duplicate data, checking for inconsistent data and correcting errors in the data. The following is an image of the results of preprocessing the learning outcomes dataset in the WEKA application.

JURNAL VOKASI KETEKNIKAN
Vol.2, No. 2, September 2024
e-ISSN: 2986-9226
https://ojs.unm.ac.id/vokasiketeknikan

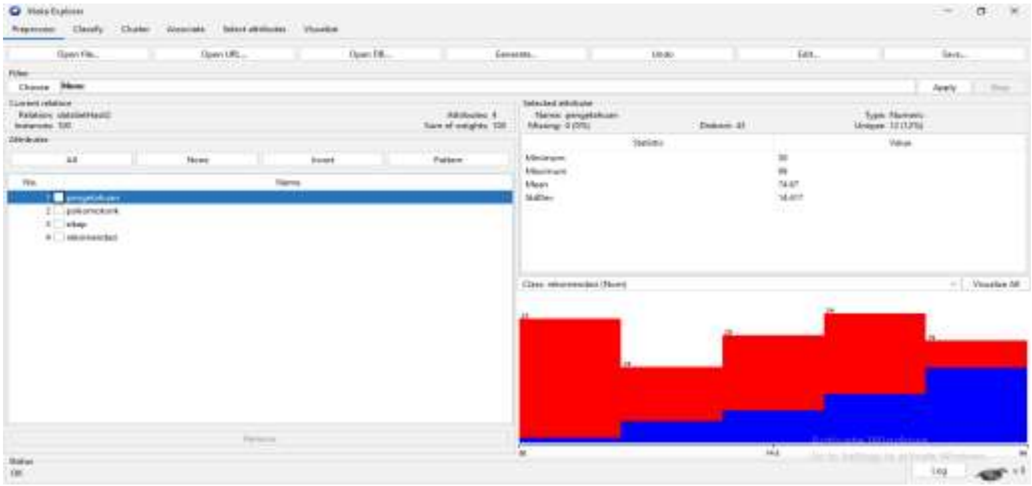
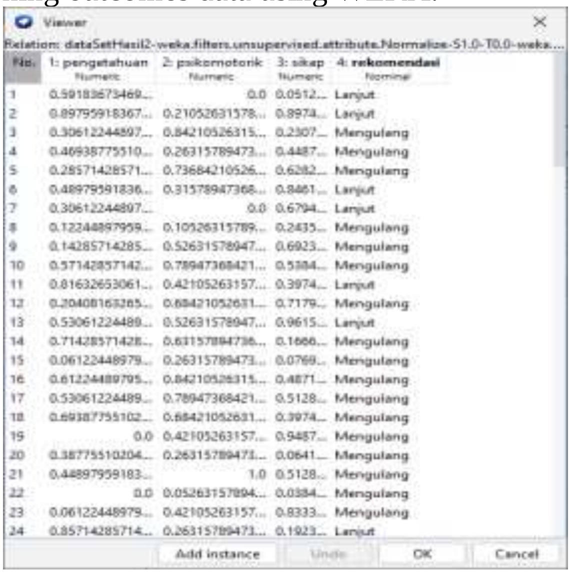


Figure 2. Results of data preprocessing with WEKA

3. Transformation

At this stage, the size of the attribute values in the learning outcomes dataset, which previously had different scales (the psychomotor attribute scale is different from the knowledge and attitude attribute scale), is standardized. The scale of each attribute is then normalized to between the values 0 and 1. Data normalization aims to reduce data skew, increase algorithm convergence, improve numerical stability, and optimize model performance (Singh et al., 2021), (Shanker et al., 2020), (Zhao et al., 2019), (Garcia et al., 2022), (Patel et al., 2020). The following are the results of normalizing learning outcomes data using WEKA.



Viewer
Relation: dataSetHasil2-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0-weka...

No.	1: pengetahuan Numeric	2: psikomotorik Numeric	3: sikap Numeric	4: rekomendasi Nominal
1	0.59183673469...	0.0	0.0512...	Lanjut
2	0.89795918367...	0.21052631578...	0.8974...	Lanjut
3	0.30612244897...	0.84210526315...	0.2307...	Mengulang
4	0.46938775510...	0.26315789473...	0.4487...	Mengulang
5	0.28571428571...	0.73684210526...	0.6282...	Mengulang
6	0.48979591836...	0.31578947368...	0.8461...	Lanjut
7	0.30612244897...	0.0	0.6794...	Lanjut
8	0.12244897959...	0.10526315789...	0.2435...	Mengulang
9	0.14285714285...	0.52631578947...	0.6923...	Mengulang
10	0.57142857142...	0.78947368421...	0.5384...	Mengulang
11	0.81632653061...	0.42105263157...	0.3974...	Lanjut
12	0.20408163265...	0.68421052631...	0.7179...	Mengulang
13	0.53061224489...	0.52631578947...	0.9615...	Lanjut
14	0.71428571428...	0.63157894736...	0.1666...	Mengulang
15	0.06122448979...	0.26315789473...	0.0769...	Mengulang
16	0.61224489795...	0.84210526315...	0.4871...	Mengulang
17	0.53061224489...	0.78947368421...	0.5128...	Mengulang
18	0.69387755102...	0.68421052631...	0.3974...	Mengulang
19	0.0	0.42105263157...	0.9487...	Mengulang
20	0.38775510204...	0.26315789473...	0.0641...	Mengulang
21	0.44897959183...	1.0	0.5128...	Mengulang
22	0.0	0.05263157894...	0.0384...	Mengulang
23	0.06122448979...	0.42105263157...	0.8333...	Mengulang
24	0.85714285714...	0.26315789473...	0.1923...	Lanjut

Add instance | Undo | OK | Cancel

Figure 3. Data normalization results with WEKA

4. Datamining

At this stage, the Naïve Bayes algorithm is applied to the dataset to solve the classification problem. In classification, 2 test options are selected for the dataset to see which test option is more accurate in classifying the data. The first test option is cross-validation with fold 10, namely 100 data in the dataset, which will be divided into two parts. The first part, namely 90 data, will be used as training data, and ten will be used as testing data. Fold validation is set to 10, as it will provide a good balance between bias and variance. By dividing the data into ten folds, we get a fairly detailed evaluation of the model's performance on various subsets, reducing the possibility of overfitting. Kohavi (1995) states that 10-fold cross-validation provides more stable and reliable error estimates than smaller or larger k-folds. Additionally, 10-fold cross-validation has become the de facto standard in many machine learning research and applications. This makes research results more accessible to compare and adopt by the scientific and practitioner communities. Witten et al. (2016), in their book "Data Mining: Practical Machine Learning Tools and Techniques", recommend using 10-fold cross-validation as standard practice in evaluating machine learning models. The second test option is percentage split, which is set to 80; the dataset is divided into two parts, where 80 per cent of the data becomes training data, and 20 per cent becomes testing data. The percentage split to 80 was calculated to balance training and testing data. According to Goodfellow et al. (2016), this proportion provides enough data for good training while leaving sufficient data to evaluate model performance. Additionally, by using 80% of the data for training, the model has enough information to learn from the data, which helps reduce the risk of overfitting. Overfitting occurs when a model is too complex and adapts to the training data to the point of losing the ability to generalize to new data. Both Hinton et al. (2012) and Srivastava et al. (2014) highlight the importance of having enough training data to build generalist models. Figure 4 and Figure 5 are the results of data mining on the learning outcomes dataset.

JURNAL VOKASI KETEKNIKAN

Vol.2, No. 2, September 2024

e-ISSN: 2986-9226

https://ojs.unm.ac.id/vokasiketeknikan

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          89               89      %
Incorrectly Classified Instances        11               11      %
Kappa statistic                          0.742
Mean absolute error                      0.2276
Root mean squared error                  0.312
Relative absolute error                 50.5191 %
Root relative squared error             65.7899 %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.735    0.030    0.926      0.735    0.820      0.752    0.942     0.918     Lanjut
                 0.970    0.265    0.877      0.970    0.921      0.752    0.942     0.965     Mengulang
Weighted Avg.    0.890    0.185    0.893      0.890    0.886      0.752    0.942     0.949

=== Confusion Matrix ===

  a  b   <-- classified as
 25  9 |  a = Lanjut
  2 64 |  b = Mengulang
```

Figure 4. Datamining results with test option cross-validation with fold = 10

```
Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          15               75      %
Incorrectly Classified Instances         5               25      %
Kappa statistic                          0.4186
Mean absolute error                      0.2967
Root mean squared error                  0.4304
Relative absolute error                 63.692  %
Root relative squared error             86.9609 %
Total Number of Instances               20

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.375    0.000    1.000      0.375    0.545      0.514    0.865     0.856     Lanjut
                 1.000    0.625    0.706      1.000    0.828      0.514    0.865     0.914     Mengulang
Weighted Avg.    0.750    0.375    0.824      0.750    0.715      0.514    0.865     0.892

=== Confusion Matrix ===

  a  b   <-- classified as
  3  5 |  a = Lanjut
  0 12 |  b = Mengulang
```

Figure 5. Datamining results with test option percentage split 80

5. Interpretation/Evaluation

This stage is the final stage in KDD, namely carrying out the process of interpreting or evaluating data mining results to see the accuracy of the classification results using the Naïve Bayes algorithm. One way to measure the accuracy of a machine learning algorithm is to use a Confusion Matrix. Confusion matrix is a tool for analyzing how good a classification is. True positive (TP) and actual negative (TN) provide results when the classification is correct, while false positive (FP) and false negative (FN) provide results when the classification is incorrect. Figures 6 and 7 show the results of the confusion matrix.

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall
                 0.735    0.030    0.926      0.735
                 0.970    0.265    0.877      0.970
Weighted Avg.    0.890    0.185    0.893      0.890

=== Confusion Matrix ===

  a  b   <-- classified as
 25  9 |  a = Lanjut
  2 64 |  b = Mengulang
```

Figure 6. Confusion matrix results for test option cross-validation fold = 10

```
                 TP Rate  FP Rate  Precision  Recall
                 0.375    0.000    1.000      0.375
                 1.000    0.625    0.706      1.000
Weighted Avg.    0.750    0.375    0.824      0.750

=== Confusion Matrix ===

  a  b   <-- classified as
  3  5 |  a = Lanjut
  0 12 |  b = Mengulang
```

Figure 7. Confusion matrix results for test option percentage split = 80

Figure 6 shows that a is the label for Continue and b for Repeat. The TP is indicated by the number 25, FP is indicated by the number 2, FN is indicated by the number 9, and TN is indicated by the number 64. Figures 8 and 9 show the classification accuracy results using the Naïve Bayes algorithm.

```
Correctly Classified Instances          89               89      %
Incorrectly Classified Instances        11               11      %
Kappa statistic                          0.742
Mean absolute error                      0.2276
Root mean squared error                  0.312
Relative absolute error                 50.5191 %
Root relative squared error             65.7899 %
Total Number of Instances              100
```

Figure 8. Accuracy results of naïve Bayes classification with cross-validation fold = 10

```
Correctly Classified Instances          15               75      %
Incorrectly Classified Instances         5               25      %
Kappa statistic                          0.4186
Mean absolute error                      0.2967
Root mean squared error                  0.4304
Relative absolute error                 63.692  %
Root relative squared error             86.9609 %
Total Number of Instances               20
```

Figure 9. Accuracy results of naïve Bayes classification with percentage split = 80

JURNAL VOKASI KETEKNIKAN
Vol.2, No. 2, September 2024
e-ISSN: 2986-9226
https://ojs.unm.ac.id/vokasiketeknikan

From the classification using the naïve Bayes algorithm, an accuracy rate of 89% was obtained, precision = 0.92 and recall = 0.73, if using cross-validation with fold = 10, and an accuracy rate of 75%, precision = 1, and recall = 0.37, if using percentage split = 80. The accuracy results are obtained with the equation:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \qquad (1)$$

As for precision and recall, it is obtained using the following equation:

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

Based on the results of calculating the accuracy, precision, and sensitivity of the naïve Bayes algorithm with both cross-validation fold = 10 and percentage split = 80, the naïve Bayes algorithm can classify and provide recommendations for the system to decide whether students who are studying using the ITSJava application can continue to the next material or repeat the material being studied.

D.CONCLUSION

Based on the results obtained, it can be concluded that:

The naïve Bayes method is accurate and precise in avoiding false positives, although it is less sensitive in capturing all positive samples.Applying the Naïve Bayes method to the ITS Java application can determine the level of students' ability to understand the material. Cross-validation with fold = 10 is the test option suitable for carrying out the data mining process using the naïve Bayes algorithm. The model for determining student ability to understand material using Naïve Bayes may need to be adjusted or improved by increasing recall without significantly sacrificing precision. This can be done using techniques such as data balancing, setting thresholds, or combining the model with other techniques that can address recall weaknesses.

REFERENCES

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (2020). Cognitive tutors: Lessons learned. Journal of the Learning Sciences, 29(4), 474-512.

Baker, R. S., & Siemens, G. (2018). Educational data mining and learning analytics: Potentials and possibilities. British Journal of Educational Technology, 49(4), 537-553.

Garcia, M. A., Herrero, A., & Corchado, E. (2022). Enhancing the robustness of machine learning models with data normalization techniques. Pattern Recognition Letters, 145, 1-8.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

JURNAL VOKASI KETEKNIKAN
Vol.2, No. 2, September 2024
e-ISSN: 2986-9226
https://ojs.unm.ac.id/vokasiketeknikan

Hinton, G. E., Osindero, S., & Teh, Y. W. (2012). A fast learning algorithm for deep belief nets. Neural Computation, 18(7), 1527-1554.

Khosravi, H., Sadiq, S., & Gasevic, D. (2022). Supporting personalized learning in intelligent tutoring systems with machine learning. Artificial Intelligence in Education, 32(2), 123-143.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 2, 1137-1143.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2019). A Bayesian Network approach to modeling the assessment of complex learning. Educational Measurement: Issues and Practice, 38(2), 38-47.

Nguyen, T., Gardner, L. A., & Sheridan, D. (2020). A multi-layered approach for improved student engagement in online learning: A case study of Deakin University, Australia. Computers & Education, 144, 103706.

Patel, H., Prajapati, M., & Shah, M. (2020). A comparative study on data normalization techniques for machine learning. Journal of Statistics and Management Systems, 23(2), 181-200.

Pebdika, A., Herdiana, R., Solahudin, D., (2023). Klasifikasi Menggunakan Metode Naïve Bayes untuk Menentukan Calon Penerima PIP. Jurnal Mahasiswa Teknik Informatika, 7(1), 452-458.

Shanker, S., Balakrishnan, S., & Shanker, M. (2020). Effect of data normalization on deep neural network training. International Journal of Applied Engineering Research, 15(7), 1876-1884.

Singh, S., Garg, H., & Goyal, P. (2021). Data normalization: A preprocessing step for machine learning. Journal of Data Science, 19(3), 356-371.

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

Wang, Y., Heffernan, N. T., & Heffernan, C. (2021). Using decision trees to model the knowledge state of students in intelligent tutoring systems. Educational Technology & Society, 24(2), 113-124.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.

Woolf, B. P. (2019). Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. Morgan Kaufmann.

Zhao, Y., Wang, G., & Li, J. (2019). The importance of data preprocessing in machine learning: A case study. IEEE Access, 7, 123967-123976.

Zhou, Z., Zhang, Y., & Li, C. (2021). Intelligent tutoring systems and the development of educational technology. Journal of Educational Technology & Society, 24(1), 16-29.