

Classification Algorithms to Determine Students' Specialization in a Higher Education Institution

Tri Basuki Kurniawan^{1*}, Indah Hidayanti¹

¹Magister Program, Universitas Bina Darma, Palembang, Indonesia

*Email: tribasukikurniawan@binadarma.ac.id

Abstract

One of the higher education institutions, namely the Faculty of Computer Science of Bina Darma University in Palembang offers courses in information technology (IT). Database, software, and network infrastructure are the areas of specialization available through the IT Study Program at the Faculty of Computer Science. These courses are complementary to those offered at Bina Darma University. Those areas of specialization must be chosen in the fourth or fifth semester, however, many students are still confused and unaware of their interests and potential which may lead to choosing a specialization that does not suit them. In this view, students may not be graduating on time. The study in this article is inspired by this situation. Our idea is to present a prediction model that assists faculty in identifying the best specialization for each student. Primary datasets are those that were gathered from the faculty and include 3599 records with 42 attributes. After that, we looked at how Python programming classification algorithms like Support Vector Machine (SVM), Naïve Bayes, Random Forest, and Decision Tree performed in classifying the areas of specialization of the students. This study demonstrates that the Decision Tree and Naïve Bayes programs reach high accuracy rates of 98,06% and 92,78%, respectively.

Keywords

Decision Tree, Naïve Bayes, Random Forest, SVM, Classification Algorithms

Introduction

The number of universities will grow along with the number of high-quality human resources these universities create. The percentage of students who can finish their coursework on time is one element that defines the quality of higher education. Naturally, this is accomplished by matching the student's academic aptitude with the appropriate focus throughout the lecture session. Determining a student's focus requirements is essential to determining their interests and skills. Students are believed to be able to graduate on schedule by selecting the appropriate concentration.

In particular, the Faculty of Computer Science at Bina Darma University Palembang offers an Information Technology (IT) Study Program. The IT Study Program's specialization in areas of competence such as database management, software development, and network infrastructure

Submission: 6 December 2023; **Acceptance:** 13 December 2023



Copyright: © 2023. All the authors listed in this paper. The distribution, reproduction, and any other usage of the content of this paper is permitted, with credit given to all the author(s) and copyright owner(s) in accordance to common academic practice. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license, as stated in the website: <https://creativecommons.org/licenses/by/4.0/>

align with Bina Darma University's IT Study Program curriculum. Ideal selection criteria for the skill concentration method consider the academic aptitude and interests of the students. Students must select from among the available significant concentrations. Concentration is determined after the fourth semester. On the other hand, students often lack awareness of their interests and skills. Students are currently having trouble choosing their significant concentration. It is believed that preferences are necessary to assist students in selecting a focus because selecting a considerable concentration is solely dependent on the student's desires or what their peers recommend. It is intended that this will serve as a guide for students as they choose their following areas of concentration.

To determine the most accurate test results in managing student data information as a basis for determining student expertise concentration, the author of this research will compare the performance of the two classification methods in data mining, namely the Decision Tree, Naive Bayes, Random Forest, and Support Vector Machine algorithms, using data samples from the University of Bina Darma for IT Study Program. The information is derived from course grades from the first to the fourth semesters. The goal of comparing the algorithm's performances is to determine which algorithm performs at the highest level of accuracy when it comes to choosing student skill concentrations.

Many studies have been conducted previously on the classification and application of data mining. Some research on classification includes research conducted by Anam & Santoso (2018), namely a comparison of the performance of the C4.5 and Naïve Bayes algorithms for classifying scholarship admissions. The study uses secondary data in the form of a list of scholarship applicants and recipients as a data set, which has six factors' determinants, namely semester, GPA, co-/extra-curricular achievements, parents' income, electricity costs, and the number of parents' dependents—testing with 10-fold cross-validation as well as evaluating model performance using the Rapid Miner tool. The research results show that the accuracy level of the C4.5 algorithm model is 96.40%, which is better than the accuracy level of the Naive Bayes algorithm model of 95.11%.

Additionally, Yaqin (2019) studied four methods of comparative classification—K-NN, Neural Network, C.4.5, and Naïve Bayes—to determine majors. The student's NIM, name, birthdate, place of education, results from national exams, and significance of choice are the attributes that are used. The Cumulative Achievement Index for both first semester and 2nd semester will be processed together with additional data. According to study findings, the accuracy of the Naive Bayes Classifier is 73.94%, the K-NN is 54.61%, the Neural Network is 15.82%, and C.45 is 44.28%.

Other researchers compare different classification algorithms; for example, Subarkah et al. (2023) compare Naïve Bayes, Support Vector Machine (SVM), and Random Forest. Their research shows that the SVM gives a higher accuracy, 87%, in predicting student graduation. Misinem et al. (2022), Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine to predict sentiment analysis for Mobile Banking Apps in Malaysia. The result shows that the Decision Tree gives the higher accuracy, at 94.37%, followed by the Random Forest, with an accuracy of 93.69%.

Methodology

Research Stages

In this research, the dataset comes from alumni and active Informatics Engineering students. The value data were processed using several data mining methods. The following are the stages in conducting data mining research:

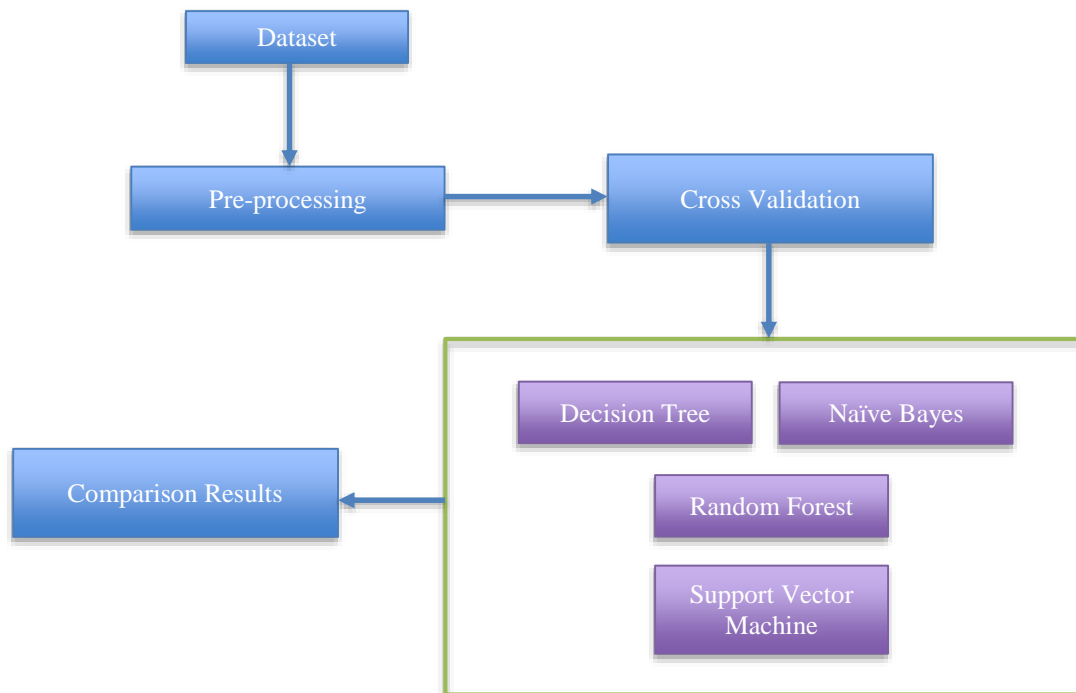


Figure 1. Research Stages

Data Collection

To obtain research data, the author collected alum data from Informatics Engineering students from the 2013 - 2014 class and data from still active students, namely the 2015-2016 class. The data taken is data that has the same curriculum and courses. The data used includes NIM data, GPA from semester 1 to semester 4, GPA in semester 4, and courses. The data was taken at the Directorate of Information Technology Systems at Bina Darma University. The data obtained was 3,599 by pre-processing, so the data deemed unfit was deleted so that the data suitable for use was 3142.

Data Mining Processing

Data mining is a subset of Knowledge Discovery in Databases (KDD), which applies a particular method to extract models or patterns from data. The following is the KDD procedure (Gullo, 2015):

Data Selection

Data was obtained from alum data from Informatic Technology students from the 2013-2014 class, and data from still active students, namely from the 2015-2016 class at the Faculty of Computer Science, Bina Darma University, which got 3142 data, for training and testing data which has a concentration in Databases, Software, and Infrastructure Networks for the 2013 to 2016 class where the data needed is course grade from semester 1 to semester 4. The data selected is from students with a GPA above 3 because they are considered appropriate and show that they choose a good concentration.

Data Pre-processing

From the data collection process, 3599 data were obtained from students who had graduated and students active with 42 attributes. However, not all data and features can be used based on their low gain values and our observation. Only thirty-two (32) significant features can be used in the classification process. The concentration is the label. Meanwhile, course attributes such as Algorithms and Programming, English 1 and 3, Business Processes, Data Structure and Data Structures Practicum, Database practicum, Discrete Mathematics, Relation Management System, and Indonesian are not used.

Several data pre-processing techniques used are (Bhaya, 2017):

- Data cleaning is used to identify and remove inconsistent data and incomplete data. In this study, students' records with the status of quitting or inactive were deleted because they contained incomplete course grades. The kind of data that is not suitable is deleted. So, the initial data was 3599, and then 457 data had to be deleted so that 3142 data would be used in the data mining process.
- Data reduction is used to obtain data sets with a smaller number of attributes and records that are informative. In this research, features like student name, student ID, and gender are not used in the mining process.

Transformation

At this stage, the data is converted into a form suitable for mining. The data transformer technique is used to make changes to data. The data transformation technique is discrete, which changes continuous type data into discrete type data. Discretize is one of the methods used when pre-processing input. Data discretization techniques can reduce the number of numeric attribute values by dividing the attribute range into intervals. Interval labels can then be used to replace actual data values. Replacing many continuous attribute values with a small number of interval labels can reduce and simplify the original data so that the process carried out is short and makes it easy to present the level of knowledge from the results of the data mining process (Jing et al., 2018).

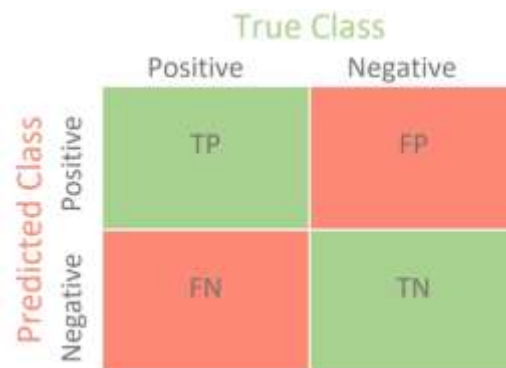
Data Mining

Data mining is a study that includes collecting, cleaning, processing, and analyzing data collection so that a deep understanding of the data can be obtained with these activities. Data mining is looking for interesting patterns or information in selected data using specific techniques or methods after carrying out the preprocessing and data transformation processes suitable for using classification data mining techniques such as the Decision Tree, Naive Bayes, Random Forest, and Support Vector Machine algorithms. The next stage is to carry out the data mining analysis process that has been carried out in the previous step.

Classification of student significant concentration based on student data using the Decision Tree, Naive Bayes, Random Forest, and Support Vector Machine algorithms based on Python programming is conducted, and the results are compared. The modeling for each classification algorithm has been created based on training data. Next, the models were tested with testing data, and the accuracy obtained in the next chapter will be compared and analyzed.

Confusion Matrix

Confusion matrix is usually used to calculate accuracy in data mining concepts. This formula performs calculations with three (3) outputs: accuracy, recall, and precision (Powers & Ailab, 2011). Confusion Matrix helps analyze the classifier's ability to recognize tuples from existing classes. The Confusion Matrix method presents the model evaluation results using a matrix table. Suppose the data set consists of two categories; the first is considered positive, and the second is considered harmful. Evaluation using the Confusion Matrix produces accuracy, precision, and recall values. An example of a Confusion Matrix can be seen in Figure 2 below:



The diagram shows a 2x2 matrix for binary classification. The columns are labeled 'True Class' with 'Positive' and 'Negative'. The rows are labeled 'Predicted Class' with 'Positive' and 'Negative'. The cells contain: TP (True Positive) in the top-left, FP (False Positive) in the top-right, FN (False Negative) in the bottom-left, and TN (True Negative) in the bottom-right. The TP and TN cells are green, while the FP and FN cells are red.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 2. Confusion Matrix for Binary Classification

The percentage of positive classes (true positive and false positive) obtained in the classification was also calculated. Recall precision functions to calculate the rate of false positives and false negatives to find the information.

Cross-Validation

One statistical technique for estimating machine learning models' competence is cross-validation. Applied machine learning is frequently used to compare and choose a model for a specific predictive modeling problem. It is simple to comprehend and apply and yields skill estimates that are typically less biased than those from other methods (Kohavi, 2001).

You want to know if your machine learning model fits the data if you have some data and a model. Your data can be divided into training and test sets. Use the training set to train your model and the test set to assess the outcome. However, you only tested the model once, so you're unsure if your favorable outcome coincided. To increase your confidence in the model's design, you should assess the model more than once.

The process takes a single parameter, k , which is the number of groups into which a given data sample should be divided. As such, k -fold cross-validation is a common name for the process. When a particular number for k is selected, it can be substituted for k in the model's reference; for example, $k=10$ would become a 10-fold cross-validation (Jung, 2017).

In applied machine learning, cross-validation is mainly used to gauge a machine learning model's proficiency in hypothetical data. That is, to assess the model's projected overall performance using a small sample size when making predictions on data that was not used for model training. It is a well-liked strategy because, compared to other approaches, such as a straightforward train/test split, it typically yields a less biased or optimistic estimate of the model performance and is easy to understand. Remember that k -fold cross-validation assesses the model's design rather than specific training because you used various training sets to retrain the model with the same design.

Results and Discussion

The implementation of the data processing process uses data discretization, which is used to reduce the number of numerical attribute values by dividing the attribute range into intervals. Interval labels can then be used to replace actual data values. K -fold cross-validation is carried out to obtain accurate results. The modeling results processed by Python programming, apart from producing modeling patterns, can also determine the accuracy, recall, and precision level. The results of a comparison between a few classification algorithms were conducted and discussed.

Split The Dataset

Assigning seventy percent of the data points to the training set and the remaining one to the testing set is the most straightforward method of dividing the modeling dataset. As a result, we use the training set to train the model before applying it to the test set. We can assess our model's performance in this way. The Python code can be seen in Figure 3.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import KBinsDiscretizer

data_url = "/dataset/clean_data.csv"
df = pd.read_csv(data_url)

display(df)

# split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 3. Python code to split the dataset into training and testing data

The distribution of data based on their labels can be seen in Figure 4 and Figure 5 below:

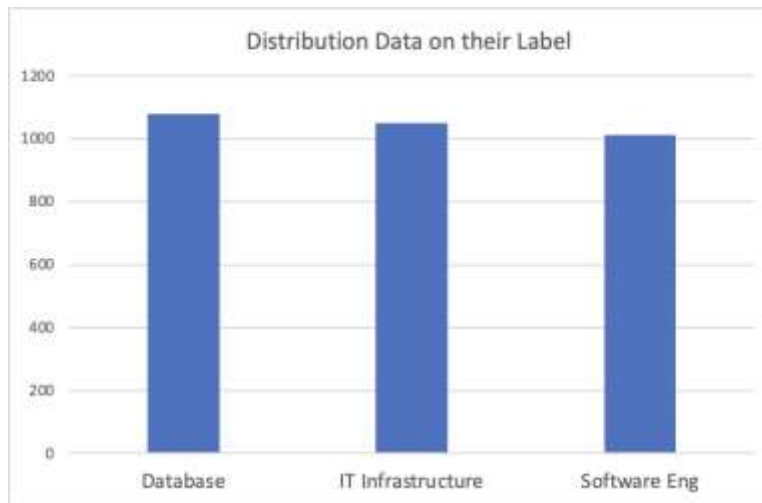


Figure 4 Distribution Data on their Label



Figure 5. Distribution of data for training and testing

Based on Figure 4, the distribution data is balanced between three (3) labels. After splitting the data, as shown in Figure 5, the distribution is still balanced. We hope the algorithms can be learned properly with the balanced dataset.

Discretization

The continuous variable is sorted into intervals with an equal number of observations using equal-frequency discretization (Putri et al., 2023). Quantiles determine the interval width. Since equal-frequency discretization distributes observations evenly throughout the various bins, it is beneficial for skewed variables.

With Scikit-learn, we may apply equal-frequency discretization, as shown in Figure 6.

```
disc = KBinsDiscretizer(n_bins=10, encode='ordinal', strategy='quantile')
disc.fit(X_train[variables])
train_t = X_train.copy()
test_t = X_test.copy()
train_t[variables] = disc.transform(X_train[variables])
test_t[variables] = disc.transform(X_test[variables])
```

Figure 6. Python code for equal-frequency discretization

Classification Modeling

Based on their training data, each model is training. Figure 7 shows the Python code for the decision tree. Using different libraries for each model algorithm, all models are created, and then validation with training and testing data is conducted. In the second part of the code, a cross-validation is performed with k being equal to 10. The results are shown in Figure 8.

```
from sklearn import tree
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_validate

dTree = tree.DecisionTreeClassifier()
dTree.fit(X_train, y_train)
print('Accuracy of Decision Tree classifier on training set: {:.2f} %'
      .format(dTree.score(X_train, y_train) * 100))
print('Accuracy of Decision Tree classifier on test set: {:.2f} %'
      .format(dTree.score(X_test, y_test) * 100))

y_pred = dTree.predict(y_test)
confusion_matrix(y_test, y_pred)

cv_results = cross_validate(dTree, X, y, cv=10)
sorted(cv_results.keys())

print('test_score Decision Tree ')
print(cv_results['test_score'])
print()

print('Maximum value', round(max(cv_results['test_score']) * 100, 2), "%")
print('Average value', round(sum(cv_results['test_score'])/len(cv_results['test_score']) * 100, 2), "%")
print()
```

Figure 7. The Python code for the decision tree algorithm

```
Accuracy of Decision Tree classifier on training set: 99.22 %
Accuracy of Decision Tree classifier on testing set: 98.09 %
[[320  2  2]
 [ 4 307  4]
 [ 2  4 297]]

test_score Decision Tree
[0.95860550 0.96160550 0.96820550 0.97360550 0.97460550 0.99030550
 0.97160550 0.97260550 0.96260550 0.95960550]

Maximum value 99.58 %
Minimum value 95.86 %
```

Figure 8. Result for the decision tree algorithm

Comparison Results

Results for each model based on their algorithms are shown in Table 1.

Table 1. Comparison results

No	Classifier Algorithms	Accuracy (%)		Cross Validation (cv=10)	
		Training	Testing	Maximum	Minimum
1	Decision Tree	99.22	98.09	99.58	95.86
2	Naïve Bayes	98.83	92.78	98.94	93.42
3	Random Forest	83.86	68.00	84.82	67.20
4	Support Vector Machine	69.53	66.99	71.02	68.26

Table 1 shows that the Decision Tree algorithm had the highest accuracy on testing data (unseen data) at 98.09%. The Support Vector Machine obtains the worst accuracy at 66.99. Overall, the testing data is lower than the training data, but the difference is only small. It means our algorithms are not overfitting. The cross-validation results show the accuracy for training and testing data are in the range of cross-validation maximum and minimum range value. It means our result did not show an anomaly or too big a difference result.

Based on their comparison results, the Decision Tree and Naïve Bayes algorithms perform quite well. This is because the Decision Tree has a mechanism in its procedure to ignore or skip the not important features. That accuracy result shows that not all features correlate well with the labels. Although the simple feature selection based on gain information and manual observation has already been applied, the accuracy result for Random Forest and SVM still shows a lower value. It is recommended to use full feature selection to apply.

Conclusion

In this study, data that has been processed using the KDD (Knowledge Discovery in Database) approach was used to model utilizing the Decision Tree, Naïve Bayes, Random Forest, and Support Vector Machine algorithms. Using the categorization method and Python programming algorithms, observations were made on student datasets from the Faculty of Information Technology at Bina Darma University. The models were developed based on the use of KDD techniques, and the outcomes indicate that while some algorithms produced good accuracy, the remaining ones produced the worst. When tested with unseen data, the Decision Tree algorithm had the highest accuracy. The accuracy obtained by the Support Vector Machine is the lowest. Although there is not a significant difference, the testing data is generally lower than the training data. Therefore, our algorithms do not exhibit overfitting. The accuracy result demonstrates that not every feature has a strong correlation with the labels. Even after applying the straightforward feature selection method based on gain information and human observation, the accuracy result for SVM and Random Forest is still lower. Applying with full feature selection is advised.

References

- Anam, C., & Santoso, H. B. (2018). *Perbandingan Kinerja Algoritma C4.5 dan Naive Bayes untuk Klasifikasi Penerima Beasiswa* (Vol. 8, Issue 1).
- Bhaya, W. (2017). Review of Data Preprocessing Techniques in Data Mining. *Journal of Engineering and Applied Sciences*, 12, 4102–4107. <https://doi.org/10.3923/jeasci.2017.4102.4107>
- Gullo, F. (2015). From Patterns in Data to Knowledge Discovery: What Data Mining Can Do. *Physics Procedia*, 62, 18–22. <https://doi.org/10.1016/j.phpro.2015.02.005>
- Jing, Y., Li, T., Fujita, H., Wang, B., & Cheng, N. (2018). An incremental attribute reduction method for dynamic data mining. *Information Sciences*, 465. <https://doi.org/10.1016/j.ins.2018.07.001>
- Jung, Y. (2017). Multiple predicting K -fold cross-validation for model selection. *Journal of Nonparametric Statistics*, 30, 1–19. <https://doi.org/10.1080/10485252.2017.1404598>
- Kohavi, R. (2001). *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. 14.
- Misinem, Kurniawan, T. B., Zaki Zakaria, M., & Aqil Azfar Bin Uzailee, M. (2022). Sentiment Analysis on Users' Satisfaction for Mobile Banking Apps in Malaysia. *JOURNAL OF DATA SCIENCE*, 2022(11). <http://ipublishing.intimal.edu.my/jods.html>
- Powers, D., & Ailab. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *J. Mach. Learn. Technol*, 2, 2229–3981. <https://doi.org/10.9735/2229-3981>
- Putri, P., Prasetyowati, S., & Sibaroni, Y. (2023). The Performance of the Equal-Width and Equal-Frequency Discretization Methods on Data Features in Classification Process. *Sinkron*, 8, 2082–2098. <https://doi.org/10.33395/sinkron.v8i4.12730>
- Subarkah, A., Kusumawati, R., & Imamudin, M. (2023). Comparison of Different Classification Techniques to Predict Student Graduation. *MATICS: Jurnal Ilmu Komputer Dan Teknologi Informasi (Journal of Computer Science and Information Technology)*, 15, 96–101. <https://doi.org/10.18860/mat.v15i2.24095>
- Yaqin, Moh. A. (2019). *Komparasi Metode Klasifikasi Dalam Penentuan Penjurusan Dengan Menggunakan 4 Metode (K-NN, Neural Network, C.4.5 Dan Naive Bayes)*. <https://api.semanticscholar.org/CorpusID:202145214>