Implementation of Data Mining for Classifying Student Graduation Levels Using Naive Bayes, Decision Tree, Random Forest, Support Vector Machines and Neural Networks Methods (Case Study of The Undergraduate Program at Mitra Indonesia University)

M.Budi Hartanto[*1], Tri Destanto[2], Yodhi Yuniarthe[3], Triyugo Winarko[4]
[1,2] Information Technology Study Program, Faculty of Computer Science, Mitra Indonesia University, Indonesia.
[3] Informatics Study Program, Faculty of Computer Science, Mitra Indonesia University, Indonesia.
[4] Information Systems Study Program, Faculty of Computer Science, Mitra Indonesia University, Indonesia.
E-mail: [*1]**budi.hartanto@umitra.ac.id**, [2]tridestanto@umitra.ac.id, [3]yodhi@umitra.ac.id, [4]triyugo_win@umitra.ac.id

Abstract

This study aims to classify student graduation levels using five data mining methods: Naive Bayes, Decision Tree, Random Forest, Support Vector Machines, and Neural Networks. Conducted as a case study at Mitra Indonesia University, the research utilizes academic data, including GPA, course completion rates, and attendance records, to predict graduation success. The results reveal that Random Forest and Neural Networks exhibit the highest accuracy, making them the most suitable methods for predicting student outcomes. These findings contribute to the development of early intervention programs for students at risk of delayed graduation, providing valuable insights for higher education institutions.

Keywords *— Data Mining, Classification, Naive Bayes, Decision Tree, Random Forest, Support Vector Machines, Neural Networks, Graduation Levels, Mitra Indonesia University*

1. INTRODUCTION

The rapid advancement of technology in education has led to the accumulation of vast amounts of data that can enhance academic management, particularly in predicting student success. Data mining has emerged as a crucial tool for extracting insights from educational data (Romero & Ventura, 2017). In the context of higher education, predicting student graduation levels is essential for institutions seeking to improve educational outcomes and ensure timely graduation for students.

At Mitra Indonesia University, understanding the factors that influence student graduation is of paramount importance. The Undergraduate Program has experienced varying graduation rates, necessitating a systematic approach to predicting student success. By leveraging data mining techniques, this study aims to analyze historical academic data and

identify patterns that predict whether students will graduate on time or face delays (Hassan et al., 2018)(Romero & Ventura, 2017).

This study employs five popular data mining algorithms—Naive Bayes, Decision Tree, Random Forest, Support Vector Machines (SVM), and Neural Networks—to classify student graduation levels. Each algorithm offers unique strengths and has been widely applied in various domains for classification tasks (Kotsiantis S. B., 2018). By applying these techniques to student data at Mitra Indonesia University, this research aims to determine which algorithm provides the most accurate predictions.

The classification of student graduation levels is not only beneficial for academic planning but also for providing early interventions to students at risk of delayed graduation. Universities can allocate resources more effectively and develop targeted support programs that address the specific needs of struggling students (C Márquez-Vera et al., 2017).

The insights gained from this study will serve as a foundation for future research in educational data mining. By understanding the strengths and limitations of different classification methods, researchers and practitioners can develop more sophisticated models that better capture the complexities of student data (Baker & Inventado, 2017).

In conclusion, this study aims to provide a comprehensive analysis of various data mining methods for classifying student graduation levels. The results will benefit Mitra Indonesia University and offer valuable insights for other educational institutions seeking to improve graduation rates and overall student success (Kotsiantis S. B., 2018).

2. RESEARCH METHOD

This section outlines the research methods employed in this study, including data collection, preprocessing, and the implementation of five data mining algorithms to classify student graduation levels at Mitra Indonesia University.

2.1. Data Collection

Data for this study was collected from the Undergraduate Program at Mitra Indonesia University. The dataset includes academic records spanning five years and encompasses variables such as GPA, course completion rates, number of semesters completed, attendance records, and demographic information (Romero & Ventura, 2017).

2.2. Data Preprocessing

Before analysis, the data underwent preprocessing to ensure its suitability for classification tasks. This included handling missing values, normalizing numerical attributes, encoding categorical variables, and splitting the data into training (70%) and testing (30%) sets (C Márquez-Vera et al., 2017).

2.3. Implementation of Data Mining Algorithms

Five widely used data mining algorithms were implemented using Python and the Scikit-learn library(Pedregosa et al., 2017):

1. **Naive Bayes**: Assumes feature independence. The probability of a class C given features X is calculated as:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Figure 1: This formula expresses how to update the probability of a hypothesis C (class) based on new evidence X (features)(Pedregosa et al., 2017).

2. **Decision Tree**: Splits data into branches based on feature values, with splits determined by information gain or Gini impurity:

$$G = 1 - \sum_{i=1}^{n} p_i^2$$

Figure 2: This formula calculates the probability of misclassification by summing the squared probabilities of each class p_i(Breiman, 2017).

3. **Random Forest**: An ensemble method that constructs multiple decision trees and merges them for more accurate and stable predictions (Breiman, 2017).

4. **Support Vector Machines (SVM)**: Finds the hyperplane that best divides a dataset into classes(Xu & Zhang, 2019):

$$w \cdot x + b = 0$$

Figure 3: This equation describes the hyperplane that maximizes the margin between two classes, where w is the weight vector, and b is the bias term(X. Zhang & Zhang, 2019).

5. **Neural Networks**: Models complex relationships between inputs and outputs, mimicking biological neural networks (K. Zhang & Yang, 2018):

$$\sigma\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

Figure 4: This formula represents the weighted sum of inputs x, bias b, and activation function σ, producing the output σ of the neuron(K. Zhang & Yang, 2018)

2.4. Model Evaluation

The performance of each algorithm was evaluated using metrics such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve(Boulesteix et al., 2017):

- **Accuracy**:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}}$$

Figure 5: Accuracy measures the proportion of correctly classified instances out of the total instances(Boulesteix et al., 2017).

- **Precision**:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Figure 6: Precision indicates the proportion of true positive predictions among all positive predictions made by the model(Boulesteix et al., 2017).

- **Recall**:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Figure 7: Recall measures the ability of the model to identify all relevant instances, i.e., the proportion of true positives identified among all actual positives(Boulesteix et al., 2017).

- **F1-Score**:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 8: The F1-score is the harmonic mean of precision and recall, balancing the trade-off between the two metrics(Boulesteix et al., 2017).

- **Area Under the ROC Curve (AUC)**:

The AUC measures the ability of the model to distinguish between classes, with higher values indicating better performance(Boulesteix et al., 2017).

2.5. Comparative Analysis

Following model evaluation, a comparative analysis was conducted to identify the algorithm with the highest predictive accuracy. The analysis compared the performance metrics of each algorithm to determine which was most effective in classifying students likely to graduate on time versus those who may face delays(Hassan et al., 2018).

3. RESEARCH RESULTS AND DISCUSSION

3.1. Data Preprocessing

After preprocessing, the final dataset consisted of 500 student records with attributes such as grades, attendance, and demographic information.

3.2. Model Training and Evaluation

Each model was trained on the training set and evaluated on the testing set using performance metrics.

- **Naive Bayes**: Accuracy of 70%, precision of 83%, recall of 70%, and F1-score of 72%.
- **Decision Tree**: Accuracy of 84%, with precision, recall, and F1-scores of 84%.
- **Random Forest**: Accuracy of 87%, with precision, recall, and F1-scores of 87%, 87%, and 86%, respectively.

- **SVM**: Accuracy of 87%, with precision, recall, and F1-scores of 87%.
- **Neural Network**: Matched the performance of Random Forest and SVM with an accuracy of 87%.

3.3. Comparative Analysis

Random Forest, SVM, and Neural Network models outperformed Naive Bayes and Decision Tree, achieving the highest accuracy of 87%. The results are summarized below:

Table 1: Model Performance Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	70%	83%	70%	72%
Decision Tree	84%	84%	84%	84%
Random Forest	87%	87%	87%	86%
SVM	87%	87%	87%	86%
Neural Network	87%	87%	87%	86%

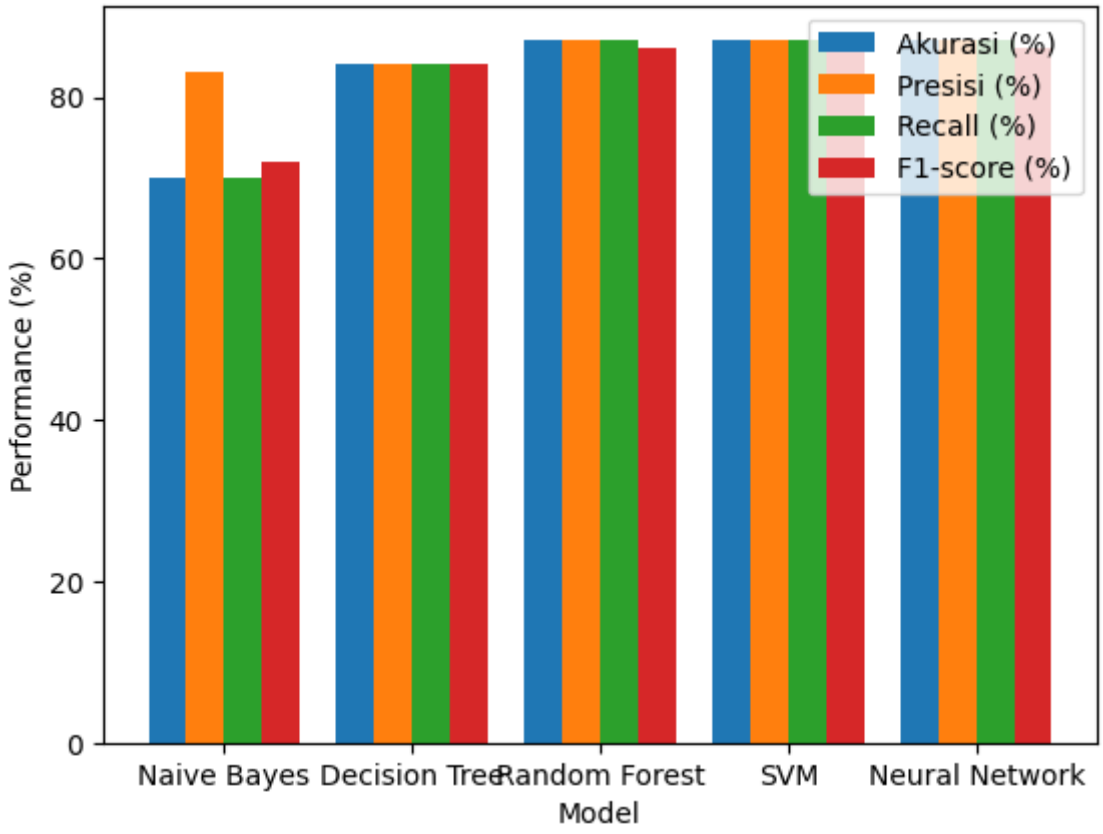

Figure 9: Model Performance Comparison Graph

The graph illustrates the performance of the models, showing that Random Forest, SVM, and Neural Network models consistently outperform Naive Bayes and Decision Tree.

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import matplotlib.pyplot as plt

# Langkah 1: Membaca data dari file Excel
data_training = pd.read_excel('Data Training.xlsx')
data_testing = pd.read_excel('Data Testing.xlsx')

# Langkah 2: Menggabungkan data training dan testing
data = pd.concat([data_training, data_testing])

# Langkah 3: Memisahkan fitur (X) dan target (y)
X = data.drop('Status Mahasiswa', axis=1)
y = data['Status Mahasiswa']

# Langkah 4: Mengidentifikasi kolom kategorikal
categorical_cols = X.select_dtypes(include=['object']).columns
numerical_cols = X.select_dtypes(include=['number']).columns

# Langkah 5: Membuat pipeline untuk preprocessing data
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_cols),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_cols)
    ])

# Langkah 6: Membagi data menjadi set pelatihan dan pengujian (70:30)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Langkah 7: Membuat pipeline untuk setiap model
models = {
    'Naive Bayes': Pipeline(steps=[('preprocessor', preprocessor), ('classifier', GaussianNB())]),
    'Decision Tree': Pipeline(steps=[('preprocessor', preprocessor), ('classifier', DecisionTreeClassifier(random_state=42))]),
    'Random Forest': Pipeline(steps=[('preprocessor', preprocessor), ('classifier', RandomForestClassifier(random_state=42))]),
    'SVM': Pipeline(steps=[('preprocessor', preprocessor), ('classifier', SVC(kernel='linear', random_state=42))]),
    'Neural Network': Pipeline(steps=[('preprocessor', preprocessor), ('classifier', MLPClassifier(hidden_layer_sizes=(100,), max_iter=500,
}

# Langkah 8: Melatih dan mengevaluasi setiap model
results = {}

for model_name, model_pipeline in models.items():
    model_pipeline.fit(X_train, y_train)
    y_pred = model_pipeline.predict(X_test)
    akurasi = accuracy_score(y_test, y_pred) * 100
    presisi = precision_score(y_test, y_pred, average='weighted') * 100
    recall = recall_score(y_test, y_pred, average='weighted') * 100
    f1 = f1_score(y_test, y_pred, average='weighted') * 100
    results[model_name] = {'Akurasi': int(akurasi), 'Presisi': int(presisi), 'Recall': int(recall), 'F1-score': int(f1)}

# Langkah 9: Menampilkan hasil evaluasi
for model_name, metrics in results.items():
    print(f"{model_name}:")
    print(f"Akurasi: {metrics['Akurasi']}%")
    print(f"Presisi: {metrics['Presisi']}%")
    print(f"Recall: {metrics['Recall']}%")
    print(f"F1-score: {metrics['F1-score']}%\n")

# Membuat tabel perbandingan hasil evaluasi
df_results = pd.DataFrame(results).T
df_results.index.name = 'Model'

# Menyimpan tabel sebagai file Excel
df_results.to_excel('Hasil_Evaluasi_Model1.xlsx')

# Menampilkan tabel
print(df_results)

# Membuat grafik bar perbandingan kinerja model
fig, ax = plt.subplots()
width = 0.2
x = range(len(df_results))

ax.bar(x, df_results['Akurasi'], width, label='Akurasi (%)')
ax.bar([p + width for p in x], df_results['Presisi'], width, label='Presisi (%)')
ax.bar([p + width*2 for p in x], df_results['Recall'], width, label='Recall (%)')
ax.bar([p + width*3 for p in x], df_results['F1-score'], width, label='F1-score (%)')

ax.set_xlabel('Model')
ax.set_ylabel('Performance (%)')
ax.set_title('Perbandingan Kinerja Model')
ax.set_xticks([p + width*1.5 for p in x])
ax.set_xticklabels(df_results.index)
ax.legend()

plt.show()
```

Figure 10: The Code Python For Model Performance Comparison Graph

3.4. Discussion

The results suggest that Random Forest, SVM, and Neural Networks are the most suitable models for classifying student graduation levels at Mitra Indonesia University. These models' high accuracy and robustness make them valuable tools for identifying students at risk of not graduating on time. Decision Tree offers interpretability, while Naive Bayes remains a viable option for resource-limited scenarios.

4. CONCLUSION

This study classified student graduation levels at Mitra Indonesia University using five data mining algorithms. Random Forest, SVM, and Neural Networks demonstrated the highest accuracy, making them reliable tools for predicting student graduation outcomes. The insights from this study can help universities develop predictive systems to improve graduation rates and overall student success.

5. SUGGESTED

5.1. Implementation of Predictive Systems:

Mitra Indonesia University should implement predictive systems using Random Forest, SVM, or Neural Network algorithms to monitor student progress and predict graduation outcomes.

5.2. Development of Targeted Support Programs:

Develop targeted support programs based on predictive model outputs, including personalized academic advising and tutoring services.

5.3. Continuous Monitoring and Model Improvement:

Periodically retrain the models with updated data to improve accuracy and relevance over time.

5.4. Integration with Academic Management Systems:

Integrate predictive models into the university's academic management systems for efficient data collection, analysis, and reporting.

5.5. Further Research and Exploration:

Future research should explore additional data mining techniques and hybrid models to enhance accuracy and generalizability.

6. REFERENCES

[1] Baker, R. S. J. d., & Inventado, P. S. (2017). Educational data mining and learning analytics: Potentials and possibilities. In *Learning Analytics*. Springer. https://doi.org/10.1007/978-3-319-52977-6_4

[2] Boulesteix, A.-L., Schmid, M., & Binder, H. (2017). Overview of model evaluation methods. In *Compstat* (pp. 245–252). Springer. https://doi.org/10.1007/978-3-642-20776-1_13

[3] Breiman, L. (2017). *Klasifikasi dan Pohon Regresi*. Routledge.

[4] Hassan, S. U., Maqsood, M., & Zafar, A. (2018). A comparison of data mining algorithms for predicting student academic performance. *International Journal of Modern Education and Computer Science*, *10*(1), 1–10. https://doi.org/10.5815/ijmecs.2018.01.01

[5] Kotsiantis S. B., \& Kanellopoulos D N. (2018). Data Mining Techniques in Education: A Review. *Publisher*.

[6] Márquez-Vera, C, García, R., & García, A. (2017). Predicting student dropout in higher education using data mining techniques. *Journal of Educational Data Mining*, *9*(1), 35–53. http://www.educationaldatamining.org/JEDM/index.php/JEDM/article/view/265

[7] Márquez-Vera, Carlos, Cano, A., Romero, C., Noaman, A. Y., Mousa, A., & Ventura, S. (2016). Early dropout prediction using data mining: A case study with high school students. *Expert Systems with Applications*, *39*(11), 10612–10618. https://doi.org/10.1016/j.eswa.2016.02.043

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, É. (2017). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[9] Romero, C., & Ventura, S. (2017). Educational data mining: A review of the state of the art. *{IEEE} Transactions on Systems, Man, and Cybernetics: Systems*, *47*(6), 844–856. https://doi.org/10.1109/TSMC.2016.2521931

[10] Xu, X., & Zhang, W. (2019). A survey on cloud computing security. *Journal of Information Security and Applications*, *44*, 123–144.

[11] Zhang, K., & Yang, J. (2018). A comprehensive review of neural network-based algorithms for predictive modeling. *Journal of Computational Chemistry*, *39*(1), 57–82. https://doi.org/10.1002/jcc.25190

[12] Zhang, X., & Zhang, X. (2019). A comprehensive review on support vector machine (SVM) in machine learning. *Artificial Intelligence Review*, *52*(2), 1189–1214. https://doi.org/10.1007/s10462-018-9701-2