The Implementation of Support Vector Machine and Naïve Bayes Algorithm to Predict Diabetes

Joshua Roy Danna Lacanlale[1*], Vitri Tundjungsari[2]

[1,2]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul, Jakarta, Indonesia
Coresponden author:
Email: jroydanna@gmail.com

Abstract.
The significant increase in diabetes mellitus cases within the community demands a technology-based solution that can provide accurate, efficient, and reliable predictions. This study aims to evaluate the impact of various data preprocessing schemes on the performance of the Gaussian Naive Bayes (GNB) and Support Vector Machine (SVM) algorithms in classifying diabetes risk. The dataset used in this research was sourced from the UCI Machine Learning Repository and consists of 520 records with 16 symptom features and 1 target label. The preprocessing stages include handling missing values, encoding categorical features, normalizing numerical data using StandardScaler, balancing the dataset with the Synthetic Minority Over-sampling Technique (SMOTE), and feature selection using the SelectKBest method. A total of nine preprocessing scheme combinations were tested for each algorithm. The experimental results show that for the GNB model, the best performance was achieved using the combination of StandardScaler, SMOTE, and SelectKBest (k=5), reaching an accuracy of 94.53%, precision 98.36%, recall 90.91%, and f1-score 94.49%. Meanwhile, for the SVM model, the highest performance was obtained through the combination of StandardScaler and RBF kernel hyperparameter tuning, achieving an accuracy of 99.04%, precision 99.05%, recall 99.04%, and f1-score 99.03%. The evaluation was conducted using metrics such as accuracy, precision, recall, F1-score, confusion matrix, and learning curve visualization. These findings highlight the critical role of proper preprocessing in enhancing predictive model performance. This study is expected to serve as a reference for developing early detection systems for diabetes based on machine learning.

keywords: *Diabetes, Classification, Preprocessing, Machine Learning, Support Vector Machine and Naïve Bayes.*

I. INTRODUCTION

In recent decades, the prevalence of diabetes mellitus has increased significantly, making it one of the most pressing public health challenges worldwide. According to data from the IDF (International Diabetes Federation) Diabetes Atlas, in 2021, approximately 537 million adults (aged 20–79) were living with diabetes. This number is projected to rise further, reaching 783 million by 2045 [1]. Timely and effective management of diabetes is crucial to prevent serious complications such as heart disease, gangrene, and nerve damage. Although conventional therapies are currently available, more innovative approaches are needed to improve early detection and treatment of this disease.

Diabetes mellitus is a collective term for metabolic disorders characterized by chronic hyperglycemia, resulting from impaired insulin secretion, insulin resistance, or a combination of both. Insulin plays a critical role as an anabolic hormone in the metabolism of carbohydrates, fats, and proteins [2]. Diabetes is classified into several types, firstly is a Type 1 diabetes, an autoimmune condition, results from the destruction of pancreatic β-cells, leading to absolute insulin deficiency; it predominantly affects children and adolescents, with rising global incidence (3% annually) and subtypes like *Latent Autoimmune Diabetes in Adults (LADA)*. In contrast, Type 2 diabetes, accounting for 90–95% of cases, stems from insulin resistance and progressive β-cell dysfunction, often linked to obesity, aging, and comorbidities such as hypertension and dyslipidemia [2-3]. Gestational diabetes emerges during pregnancy due to transient glucose intolerance, while other specific types include rare genetic forms (e.g., *MODY*), drug-induced diabetes, and disorders affecting the exocrine pancreas (e.g., pancreatitis) or endocrine system (e.g., Cushing's syndrome) [4].

Diabetes mellitus is often diagnosed at an advanced stage, where treatment options become more limited. At this stage, the risk of serious complications such as heart disease, gangrene, and nerve damage increases. Conventional methods like BMI and blood sugar measurements are often insufficiently accurate.

These methods frequently fail to account for the multidimensional factors influencing the progression of the disease. Thus, there is an urgent need for more accurate and holistic methods. This situation highlights a gap between current conditions and the desired state, where earlier detection could significantly improve health outcomes [5].

Technology-based approaches, particularly machine learning, have shown great potential in enhancing early detection and disease prediction. Machine learning can analyze large datasets to identify patterns and risk factors that may go undetected by traditional methods. This approach has been successfully applied in various healthcare fields, including cardiovascular disease and cancer prediction [6]. This demonstrates that the use of such technology enables faster and more accurate predictions.

Previous research has demonstrated the effectiveness of machine learning in predicting various diseases. For example, machine learning algorithms is applied to predict the incidence rates of various infectious diseases in Indonesia, with results showing high prediction accuracy (94%) [7]. Second, Support Vector Machine algorithm is used to predict heart disease, producing a model with 85% accuracy [8]. Third, Linear Regression algorithm is applied to predict the likelihood of lung cancer, achieving approximately 90% accuracy [9]. Fourth, Random Forest algorithm with an accuracy 27.64% is implemented to predict the epidemiology of non-communicable diseases in community health centers [10]. Although the accuracy remains low, this study demonstrates the potential of machine learning in disease classification based on medical record data. These studies affirm that machine learning approaches can improve prediction accuracy compared to conventional methods [11].

II. METHODOLOGY

A. Data Gathering

The data used in this study is secondary data obtained from the UCI Machine Learning open-source repository under the dataset name *Early Stage Diabetes Risk Prediction*. This dataset was collected through direct questionnaires from patients at a hospital in Sylhet, Bangladesh. The dataset can be downloaded from the following webpage: https://archive.ics.uci.edu/dataset/529/early+stage+diabetes+risk+prediction+dataset. The dataset consists of 520 records and includes 16 independent variables (1 numeric variable and 15 categorical variables), with one dependent variable indicating whether a patient is positive or negative for diabetes. The variables in the dataset includes *age*, *gender*, polyuria, polydipsia, sudden weight loss, weakness, polyphagia, genital thrush, visual blurring, itching, irritability, delayed healing, partial paresis, muscle stiffness, alopecia, obesity, and class (target).

B. Data Preprocessing

Data preprocessing aims to transform raw data into a usable format while ensuring its quality before proceeding to the analysis stage in the data mining process, and it plays a crucial role in ensuring the accuracy and relevance of analytical results, because through optimal data preparation, various errors and inconsistencies in the data can be avoided, thereby preventing incorrect conclusions or misleading insights. This step serves as a fundamental foundation for the entire data analysis and data mining process [12]. The steps in this preprocessing include:

1. Checking and handling missing or empty data.
2. Encoding categorical features into numerical form for variables and the class variabel using the Label Encoder technique.
3. Scaling numerical features for *age* variable using StandardScaler and MinMaxScaler. StandardScaler is a rescaling process that transforms feature distributions to have a mean of 0 and standard deviation of 1, while MinMaxScaler is a rescaling process that transforms features to a specific range, typically from 0 to 1 [13].
4. Feature selection using the SelectKBest method with the *f_classif* score and a *k* value of 5. SelectKBest is a widely used filter-based feature selection method. Unlike model-dependent approaches, it ranks feature using statistical measures [14]. In *scikit-learn*, SelectKBest supports various univariate selection approaches, with the choice of scoring function depending on the problem type, For regression tasks, f_regression (linear regression F-test) and mutual_info_regression (mutual information) are typically

used, while classification tasks often employ chi2 (chi-square test), f_classif (ANOVA F-value), or mutual_info_classif. These functions assess feature relevance through different statistical tests—such as variance analysis or entropy estimation via k-nearest neighbors. The only other critical parameter in SelectKBest is *k*, which determines the number of top-ranked features to retain [15].

5. Handling class imbalance using the *Synthetic Minority Over-sampling Technique* (SMOTE). Imbalanced data is a common problem in machine learning, where the number of instances from one class is significantly larger than those from other classes [16]. One of the techniques to tackle this is SMOTE, it generates synthetic data for the minority classes to balance their quanitity with the majority class, and the synthetic data is created based on K-Nearest Neighbors, chosen for its ease of implementation [17].

C. Modelling

Before modelling, data splitting is performed. The data is divided into 80% training data and 20% testing data. In this stage, two machine learning algorithms are used:

1. Support Vector Machine (SVM) with RBF (Radial Basis Function) kernel. SVM is capable of performing classification tasks on datasets with both linear and non-linear characteristics. Its initial procedure involves mapping individual data instances into a multidimensional feature space, where the dimensionality corresponds to the number of input features. The core objective is to determine an optimal separating hyperplane tha partitions the data into two distinct classes [vitri, 18]. The tuning process is performed on the parameters C and gamma as shown in Table 1 below. SVM is a popular supervised learning algorithm that can be used to solve classification and regression problems and many people prefer SVM because it delivers significant accuracy with low computational power requirements [18]. The main goal of SVM is to find a hyperplane in a in an N-dimensional space (where N is the number of features) that can clearly categorize data points. The selected hyperplane is the one that correctly separates the classes, and the decision boundary is determined to maximize the margin. This approach enables better generalization and improved accuracy when classifying unseen data [18-19]. One of the most popular kernel tricks SVM is RBF which can map data from a low-dimensional space to an infinite-dimensional space and making it highly effective for handling non-linearly separable data [13].

TABLE 1. Support Vector Machine Parameters

Parameter	***Parameter Values***
C (cost)	*0.01, 0.1, 1, 10, 100*
γ (Gamma)	*0.0001, 0.001, 0.01, 0.1, 1*

The C (cost) parameter adjusts the level of regularization in the SVM model—lower values enforce stronger regularization to prevent overfitting but may result in underfitting, while higher values allow the model to fit the training data more closely but increase the risk of overfitting. Proper tuning of C helps strike a balance between bias and variance. Meanwhile, the gamma parameter influences the decision boundary's sensitivity to individual data points—higher gamma values make the model more responsive to training samples, potentially leading to overfitting, whereas lower gamma values create smoother boundaries that may underfit. Optimizing gamma ensures the model achieves the right level of complexity for generalization [20].

2. Gaussian Naïve Bayes (GNB) with var_smoothing tuning. NB is a classification technique based on Bayes' theorem, which describes the probability of an event given prior knowledge and conditions related to that event. This classifier operates under the assumptions that each feature contributes independently to the probability of a given class, even though, in practice, features within the same class may exhibit some degree of dependency [vitri, 18]. For datasets containing continuous attributes, the Naïve Bayes approach is typically realized through GNB, which assumes that the continuous values of each feature follow a gaussian (normal) distribution [21]. The parameter range is used from 10^0 to 10^9, with a total of 100 values generated logarithmically using the logspace(0, 9, num=100) function [22-23].

D. Model Evaluation

Model evaluation is conducted to assesss the model's performance using several metrics, including cross-validation (with 5-fold cross-validation), learning curves, confusion matrix, accuracy, precision, recall, and f1-score.

E. Research Schemes

For this study, 18 research schemes were developed to outline the experimental steps in the data classification process using various combinations of preprocessing methods as shown in Table 2 below. These includes feature scaling for the *age* attribute, data balancing (SMOTE), feature selection (SelectKBest with k = 5), and two classification algorithms with hyperparameter tuning. Each design represents a specific configuration tested to achieve the best performance results based on evaluation metrics such as accuracy, precision, recall, and f1-score.

TABLE 2. Research Scheme

Scheme	*Preprocessing and tuning method*	*Algorithm*
1	*Without scaling the 'age' feature*	*GNB*
2	*StandardScaler ('age' feature)*	*GNB*
3	*Min-MaxScaler ('age' feature)*	*GNB*
4	*StandardScaler ('age' feature) + Hyperparameter tuning (var_smoothing)*	*GNB*
5	*Min-MaxScaler ('age' feature) + Hyperparameter tuning (var_smoothing)*	*GNB*
6	*SMOTE + StandardScaler ('age' feature) + Hyperparameter tuning (var_smoothing)*	*GNB*
7	*SMOTE + Min-MaxScaler ('age' feature) + Hyperparameter tuning (var_smoothing)*	*GNB*
8	*SMOTE + StandardScaler ('age' feature) + SelectKBest (k = 5) + Hyperparameter tuning (var_smoothing)*	*GNB*
9	*SMOTE + Min-MaxScaler ('age' feature) + SelectKBest (k = 5) + Hyperparameter tuning (var_smoothing)*	*GNB*
10	*Without scaling the 'age' feature*	*SVM*
11	*StandardScaler ('age' feature)*	*SVM*
12	*Min-MaxScaler ('age' feature)*	*SVM*
13	*StandardScaler ('age' feature) + Hyperparameter tuning with RBF kernel (C and gamma)*	*SVM*
14	*Min-MaxScaler ('age' feature) + Hyperparameter tuning with RBF kernel (C and gamma)*	*SVM*
15	*SMOTE + StandardScaler ('age' feature) + Hyperparameter tuning with RBF kernel (C and gamma)*	*SVM*
16	*SMOTE + Min-MaxScaler ('age' feature) + Hyperparameter tuning with RBF kernel (C and gamma)*	*SVM*
17	*SMOTE + StandardScaler ('age' feature) + SelectKBest (k = 5) + Hyperparameter tuning with RBF kernel (C and gamma)*	*SVM*
18	*SMOTE + Min-MaxScaler ('age' feature) + SelectKBest (k = 5) + Hyperparameter tuning with RBF kernel (C and gamma)*	*SVM*

III. RESULTS AND DISCUSSION

A. Data preprocessing

Before the data is used to train the machine learning model, a data preprocessing stage is conducted to ensure the data is clean and properly structured for processing by the chosen machine learning algorithm. The steps in this preprocessing phase include:

1. Data cleaning

The first step in data preprocessing is data cleaning, which involves checking for missing values in the dataset. The goal is to eliminate noise from the data to ensure optimal model performance.

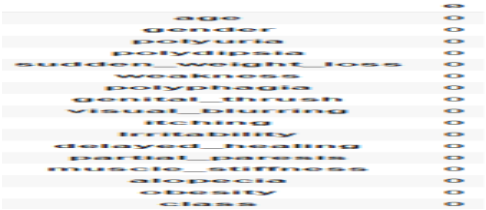

Fig. 1. Missing values cheking

As we can see from Figure 1, no missing values is found in this dataset, which means the data is free from noise.

2. Encode categorical features

After checking for missing values, encoding was performed to standardize the data types in the dataset. Figure 2 shows a total of 16 categorical features (including *class* variable) were identified, which required conversion from Object type to numerical representations and the attributes have been encoded using the Label Encoder technique, converting them into Integer (Int64) data type.

```
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   age                 520 non-null    int64
 1   gender              520 non-null    int64
 2   polyuria            520 non-null    int64
 3   polydipsia          520 non-null    int64
 4   sudden_weight_loss  520 non-null    int64
 5   weakness            520 non-null    int64
 6   polyphagia          520 non-null    int64
 7   genital_thrush      520 non-null    int64
 8   visual_blurring     520 non-null    int64
 9   itching             520 non-null    int64
 10  Irritability        520 non-null    int64
 11  delayed_healing     520 non-null    int64
 12  partial_paresis     520 non-null    int64
 13  muscle_stiffness    520 non-null    int64
 14  alopecia            520 non-null    int64
 15  obesity             520 non-null    int64
 16  class               520 non-null    int64
```

Fig. 2. Final encoded dataset for categorical features

3. Scaling numeric feature for age feature

The scaling process is performed to standardize the numerical values in the *age* feature, ensuring all values fall within a uniform range and do not dominate other features during the model training process. Large values in a feature can influence distance calculations between data points, potentially leading to a biased model [24]. Figure 3 shows two scaling techniques were used: StandardScaler and Min-MaxScaler.

	age
0	-0.661367
1	0.821362
2	-0.578993
3	-0.249498
4	0.986110

	age
0	0.324324
1	0.567568
2	0.337838
3	0.391892
4	0.594595

Fig. 3 .Numeric feature scaled using standardscaler (left) and min-maxscaler (right)

4. Feature selection using SelectKBest

The top 5 features are obtained where the selected features have the highest scores based on the f_classif evaluation, indicating the strongest correlation with the target variable, shown on Figure 4. Fig

	Specs	Score
2	polyuria	412.738410
3	polydipsia	376.422649
1	gender	130.968787
4	sudden_weight_loss	121.973731
12	partial_paresis	119.046534

Fig. 4. Five features selected based on f_classif score with k = 5

5. Synthetic Minority Over-sampling Technique (SMOTE)

Figure 5 shows that SMOTE is applied to increase the number of minority-class samples and balancing them with the majority class.

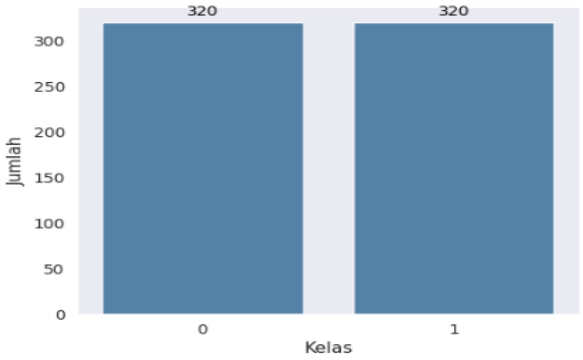

Fig. 5. Class distribution after applying SMOTE

6. Data splitting

Data split is applied in an 80:20 ratio, where 80% of the data is used for training and the remaining 20% is used for testing, shown on Table 3.

TABLE 3. Training and testing data split in 80:20 before and after SMOTE

Data	***Proportion***	***Total (without SMOTE)***	***Total (with SMOTE)***
Training	*80%*	*416*	*512*
Testing	*20%*	*104*	*128*

B. Model Evaluation

Performance comparison of schemes

The Gaussian Naïve Bayes and Support Vector Machine model was evaluated across 9 different schemes each (with total of 18 schemes), consisting of combinations of preprocessing with and without scaling, SMOTE application, feature selection, and hyperparameter tuning of each model as shown on Table 4 below.

TABLE 4. Scheme Evaluation on GNB and SVM

Scheme	***Model***	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	*GNB*	*91.35%*	*93.06%*	*94.37%*	*93.71%*
2	*GNB*	*91.35%*	*93.06%*	*94.37%*	*93.71%*
3	*GNB*	*91.35%*	*93.06%*	*94.37%*	*93.71%*
4	*GNB*	*91.35%*	*93.06%*	*94.37%*	*93.71%*
5	*GNB*	*91.35%*	*93.06%*	*94.37%*	*93.71%*
6	*GNB*	*91.41%*	*92.31%*	*90.91%*	*91.60%*
7	*GNB*	*93.75%*	*95.31%*	*92.42%*	*93.85%*
8	*GNB*	*93.75%*	*93.94%*	*93.94%*	*93.94%*
9	*GNB*	*94.53%*	*98.36%*	*90.91%*	*94.49%*
10	*SVM*	*68.27%*	*68.27%*	*100.0%*	*81.14%*
11	*SVM*	*97.12%*	*97.22%*	*98.59%*	*97.9%*

Scheme	*Model*	*Accuracy*	*Precision*	*Recall*	*F1-Score*
12	*SVM*	*91.12%*	*98.57%*	*97.18%*	*97.87%*
13	*SVM*	*99.04%*	*99.05%*	*99.04%*	*99.03%*
14	*SVM*	*97.12%*	*98.57%*	*97.18%*	*97.87%*
15	*SVM*	*97.66%*	*97.67%*	*97.66%*	*97.66%*
16	*SVM*	*98.44%*	*98.48%*	*98.44%*	*98.44%*
17	*SVM*	*92.97%*	*93.07%*	*92.97%*	*92.97%*
18	*SVM*	*93.75%*	*93.75%*	*93.75%*	*93.75%*

The Gaussian Naïve Bayes (GNB) model exhibited consistent accuracy (~91.35%) across Schemes 1–5, where only scaling variations (StandardScaler or MinMaxScaler) or no scaling of the age feature were applied. This stability reflects GNB's inherent assumption of feature independence, making it relatively robust to differences in scaling. When SMOTE and feature selection were introduced (Schemes 6–9), the model showed more variation. For instance, Scheme 9 (SMOTE + MinMaxScaler + SelectKBest) achieved the highest accuracy (94.53%) with strong precision (98.36%), while Scheme 6 (SMOTE + StandardScaler) recorded lower recall (90.91%) despite reasonable accuracy (91.41%). These results indicate that balancing and feature selection can enhance GNB performance, though specific configurations may shift the trade-off between precision and recall. In contrast, the Support Vector Machine (SVM) model was highly sensitive to preprocessing methods. Without scaling (Scheme 10), accuracy dropped drastically to 68.27%, underscoring SVM's dependency on feature scaling. Applying StandardScaler or MinMaxScaler (Schemes 11–12) significantly improved accuracy to above 91%, while hyperparameter tuning with the RBF kernel (Schemes 13–14) further boosted performance, reaching peak accuracy of 99.04%. However, when SMOTE and feature selection were combined with scaling and tuning (Schemes 15–18), performance slightly decreased (93.75–98.44%), suggesting that the RBF kernel's ability to capture non-linear patterns in high-dimensional data reduces the necessity for additional feature balancing and selection.

C. Best Scheme Evaluation of Each Model based on Confusion Matrix

Support Vector Machine

Out of the nine tested SVM schemes, scheme 4 — namely StandardScaler for the 'age' feature + Hyperparameter Tuning SVM with RBF Kernel — demonstrated the best performance with an accuracy of 99.04%, precision of 99.05%, recall of 99.04%, and F1-score of 99.03%.

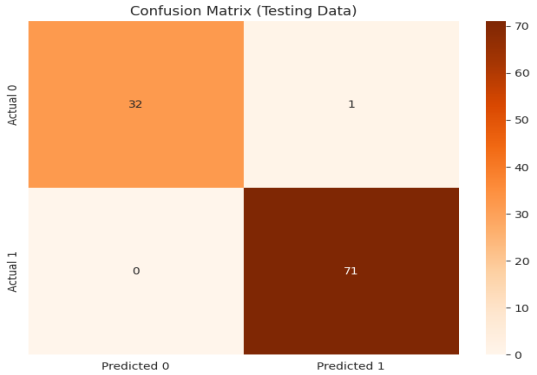

Fig. 6. Confusion matrix for the best scheme of SVM

Based on the confusion matrix shown in Figure 6, the Support Vector Machine (SVM) model tested in Scheme 4 and 5 successfully exhibited excellent classification performance. This is evident from the number of True Positives (TP) being 71 and True Negatives (TN) being 32, indicating that the majority of

the data was classified correctly. Additionally, the False Positive (FP) value was only 1, and False Negative (FN) was 0, demonstrating an extremely low classification error rate.

Gaussian Naïve Bayes

Out of the nine tested GNB (Gaussian Naive Bayes) schemes, Scheme 9 — which combines SMOTE + Min-MaxScaler for the 'age' feature + SelectKBest (k=5) + Hyperparameter Tuning for var_smoothing — showed the best performance with an accuracy of 94.53%, precision of 98.36%, recall of 90.91%, and F1-score of 94.49%.

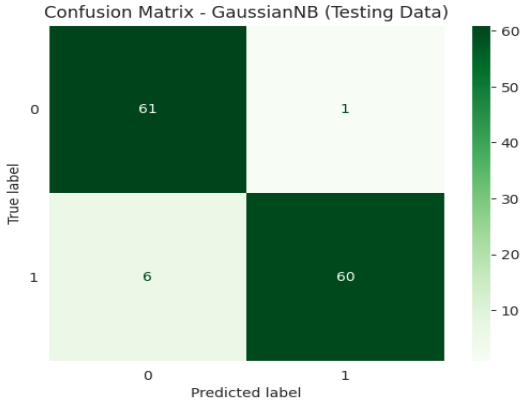

Fig 7. Confusion matrix for the best scheme of GNB

Based on the confusion matrix in Figure 7, the Gaussian Naive Bayes model tested in Scheme 8 successfully classified the test data with strong performance. There were 61 negative instances (class 0) correctly predicted (True Negative) and 60 positive instances (class 1) also classified correctly (True Positive). Meanwhile, misclassifications occurred in 1 negative instances wrongly predicted as positive (False Positive) and 6 positive instances incorrectly classified as negative (False Negative).

D. Best Scheme Evaluation of Each Model based on Learning Curves

Support Vector Machine

After evaluating performance using the confusion matrix, the next step is to assess the model's performance using learning curves.

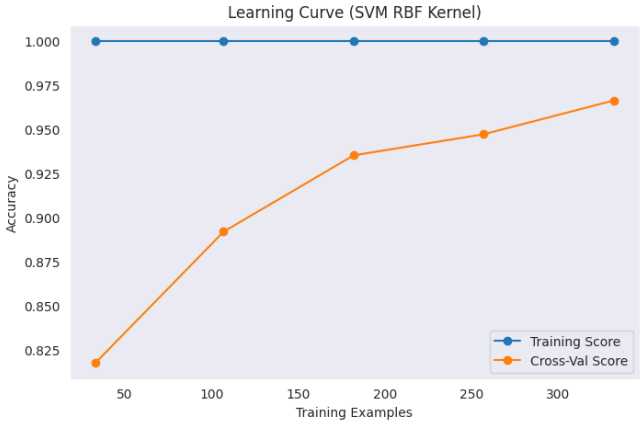

Fig. 8. Learning curve for the best scheme of SVM

From the learning curve shown in Figure 8, it can be observed that the accuracy values for the training data (training score) and cross-validation (cross-validation score) exhibit a convergent pattern as the number of training data increases. The training score remains stable at around 100%, while the cross-validation score shows a significant improvement, rising from approximately 82% to nearly 97%, with an average cross-validation accuracy of 96.63%.

Gaussian Naïve Bayes

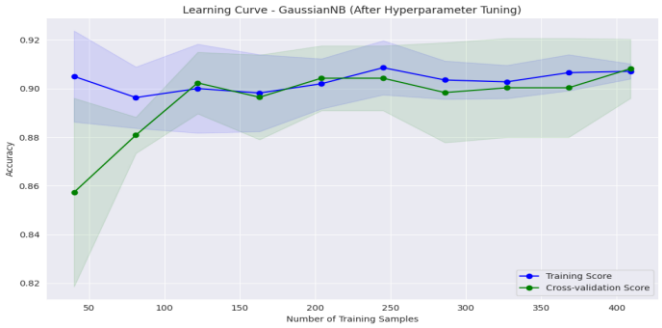

Fig. 9. Learning curve for the best scheme of GNB

From the learning curve shown in Figure 9, it can be observed that the accuracy values for the training data (training score) and cross-validation (cross-validation score) exhibit a convergent pattern as the number of training samples increases. At the beginning of the training, there is a relatively large gap between the training score and the cross-validation score, but this difference gradually narrows as the number of training samples grows. The training score remains stable within the range of 90–91%, while the cross-validation score shows a significant improvement, rising from around 85% and approaching the training score. Additionally, the average cross-validation accuracy reaches 90.81%.

IV. CONCLUSIONS

This study has evaluated the performance of Gaussian Naïve Bayes (GNB) and Support Vector Machine (SVM) algorithms in predicting diabetes mellitus, focusing on the impact of various preprocessing techniques. The results demonstrated that preprocessing significantly enhances model performance. For GNB, the optimal combination (StandardScaler, SMOTE, SelectKBest and Hyperparameter tuning) achieved an accuracy of 94.53%, precision 98.36%, recall 90.91%, and f1-score 94.49%, while SVM with StandardScaler and hyperparameter tuning with RBF kernel hyperparameter tuning reached an impressive 99.04% accuracy, 99.05% precision, 99.04% recall, and 99.03% f1-score. The findings underscore the importance of proper preprocessing, including feature scaling, class imbalance handling, and feature selection, in improving predictive accuracy. SVM outperformed GNB, particularly when optimized with hyperparameter tuning, highlighting its suitability for high-dimensional data. This study provides valuable insights for developing machine learning-based early detection systems for diabetes, emphasizing the need for tailored preprocessing to maximize model efficacy.

REFERENCES

[1] International Diabetes Federation, IDF Diabetes Atlas, 10th ed. Brussels, Belgium: International Diabetes Federation, 2021.

[2] E. Schleicher et al., "Definition, Classification and Diagnosis of Diabetes Mellitus," Experimental and Clinical Endocrinology & Diabetes, vol. 130, Apr. 2022, doi: https://doi.org/10.1055/a-1624-2897.

[3] Yohanes Denggos, "Penyakit Diabetes Mellitus Umur 40-60 Tahun di Desa Bara Batu Kecamatan Pangkep," Healthcaring, vol. 2, no. 1, pp. 55–61, Mar. 2023, doi: https://doi.org/10.47709/healthcaring.v2i1.2177.

[4] Z. Punthakee, R. Goldenberg, and P. Katz, "Definition, Classification and Diagnosis of Diabetes, Prediabetes and Metabolic Syndrome," ***Canadian Journal of Diabetes***, vol. 42, no. 1, pp. S10–S15, Apr. 2018, doi: https://doi.org/10.1016/j.jcjd.2017.10.003.

[5] K.-J. He, H. Wang, J. Xu, G. Gong, X. Liu, and H. Guan, "Global burden of type 2 diabetes mellitus from 1990 to 2021, with projections of prevalence to 2044: a systematic analysis across SDI levels for the global burden of disease study 2021," Frontiers in Endocrinology, vol. 15, Nov. 2024, doi: https://doi.org/10.3389/fendo.2024.1501690.

[6] E. Oikonomou and R. Khera, "Machine learning in precision diabetes care and cardiovascular risk prediction," Cardiovascular Diabetology, vol. 22, no. 1, Sep. 2023, doi: https://doi.org/10.1186/s12933-023-01985-3.

[7] R. G. Wardhana, G. Wang, and F. Sibuea, "Penerapan Machine Learning Dalam Prediksi Tingkat Kasus Penyakit

Di Indonesia," ***Journal of Information System Management (JOISM)***, vol. 5, no. 1, pp. 40–45, Jul. 2023, doi: https://doi.org/10.24076/joism.2023v5i1.1136.

[8] R. Hidayat, Y. S. Sy, T. Sujana, M. Husnah, H. T. Saputra, and F. Okmayura, "Implementasi Machine Learning Untuk Prediksi Penyakit Jantung Menggunakan Algoritma Support Vector Machine," BIOS : ***Jurnal Teknologi Informasi dan Rekayasa Komputer***, vol. 5, no. 2, pp. 161–168, Sep. 2024, doi: https://doi.org/10.37148/bios.v5i2.152.

[9] A. Rahman, Agung Nugroho, and A. H. Anshor, "Prediksi Penyakit Kanker Paru-Paru Dengan Algoritma Regresi Linier," Bulletin of Information Technology (BIT), vol. 4, no. 1, pp. 63–74, Mar. 2023, doi: https://doi.org/10.47065/bit.v4i1.501.

[10] F. Saptawan, David, T. Wijaya, S. Kosasi, and S. Margaretha Kuway, "Prediksi Epidemiologi Penyakit Tidak Menular Menggunakan Algoritma Random Forest Pada Puskesmas," ***Jurnal TIMES***, vol. 13, no. 2, pp. 192–201, Dec. 2024, doi: https://doi.org/10.51351/jtm.13.2.2024788.

[11] S. Basu, K. T. Johnson, and S. A. Berkowitz, "Use of Machine Learning Approaches in Clinical Epidemiological Research of Diabetes," Current Diabetes Reports, vol. 20, no. 12, Dec. 2020, doi: https://doi.org/10.1007/s11892-020-01353-5.

[12] U. N. Dulhare, K. Ahmad, Ahmad, and J. Wiley, Machine learning and big data : concepts, algorithms, tools and applications. Hoboken, Nj: Wiley-Scrivener, 2020.

[13] R. Sarno, S. Kom., Malikhah, S.Kom., M.Kom, D. Putra, and S. Hanif, Machine Learning dan Deep Learning-Konsep dan Pemrograman Python. Penerbit Andi.

[14] D. Effrosynidis and A. Arampatzis, "An evaluation of feature selection methods for environmental data," Ecological Informatics, vol. 61, p. 101224, Mar. 2021, doi: https://doi.org/10.1016/j.ecoinf.2021.101224.

[15] I. Javid, R. Ghazali, I. Syed, M. Zulqarnain, and N. A. Husaini, "Study on the Pakistan stock market using a new stock crisis prediction method," PLoS ONE, vol. 17, no. 10, p. e0275022, Oct. 2022, doi: https://doi.org/10.1371/journal.pone.0275022.

[16] Kumar Abhishek and Dr. Mounir Abdelaziz, Machine Learning for Imbalanced Data. Packt Publishing Ltd, 2023.

[17] F. R. Adi Pratama and S. I. Oktora, "Synthetic Minority Over-sampling Technique (SMOTE) for handling imbalanced data in poverty classification," ***Statistical Journal of the IAOS***, pp. 1–7, Dec. 2022, doi: https://doi.org/10.3233/sji-220080.

[18] R. Doshi, MACHINE LEARNING : master supervised and unsupervised learning algorithms with real... examples. S.L.: Bpb Publications, 2021.

[19] Aman Kharwal, Machine Learning Algorithms: Handbook. 2023.

[20] A. Ouldammar, A. Moulay Lakhdar, A. Bouida, and K. Merit, "Optimization machine learning models for selecting transmit antennas in 5G/6G systems," ***Indonesian Journal of Electrical Engineering and Computer Science***, vol. 37, no. 2, p. 819, Feb. 2025, doi: https://doi.org/10.11591/ijeecs.v37.i2.pp819-828.

[21] Yessy Asri, S.T., MMSI and Dr. Dra. Dwina Kuswardani, M.Kom, dkk, MACHINE LEARNING & DEEP LEARNING: Analisis Sentimen Menggunakan Ulasan Pengguna Aplikasi. Uwais Inspirasi indonesia, 2024.

[22] G. Vivar, R. Strobl, E. Grill, Nassir Navab, A. Zwergal, and S.-A. Ahmadi, "Using Base-ml to Learn Classification of Common Vestibular Disorders on DizzyReg Registry Data," Frontiers in Neurology, vol. 12, Aug. 2021, doi: https://doi.org/10.3389/fneur.2021.681140.

[23] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis," Informatics, vol. 8, no. 4, p. 79, Nov. 2021, doi: https://doi.org/10.3390/informatics8040079.

[24] D. U. Ozsahin, M. Taiwo Mustapha, A. S. Mubarak, Z. Said Ameen, and B. Uzun, "Impact of feature scaling on machine learning models for the diagnosis of diabetes," 2022 International Conference on Artificial Intelligence in Everything (AIE), Aug. 2022, doi: https://doi.org/10.1109/aie57029.2022.00024.