Machine Learning-Based Diabetes Classification: A Comparative Study of K-Nearest Neighbors, Support Vector Machine, Extreme Gradient Boosting, and Random Forest Techniques

1st Frederick Nathan Irmawan
Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
frederick.irmawan@binus.ac.id

2nd Nicholas Hans Muliawan
Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
nicholas.muliawan@binus.ac.id

3rd Edbert Valencio Angky
Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
edbert.angky@binus.ac.id

4th Karli Eka SetiawanComputer
Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
karli.setiawan@binus.ac.id

5th Muhammad Fikri Hasani
Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
muhammad.fikri003@binus.ac.id

***Abstract*— Diabetes is a chronic disease that influences how the human body turns food into energy and affects a significant portion of the global population. Diabetes prediction plays a major role in helping diagnose diabetes, which can help prevent complications and improve patient outcomes. This paper presents a machine learning approach for predicting the onset of diabetes using clinical and demographic data. The dataset used is from the Pima Indian Diabetes Database (PIDD), which was gathered from the National Institute of Diabetes and Digestive and Kidney Diseases. This dataset contains various variables such as pregnancies, glucose, blood pressure, skin thickness, insulin, body mass index (BMI), diabetes pedigree function, and age. Some preprocessing method, such as oversampling and removing zero values to impossible data, was done to the dataset in order to balance and improve the input data. Then the paper compare four different machine learning algorithms algorithm, which are the K-Nearest Neighbors Algorithm (KNN), Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). These models are then built to be predictive. The model's performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score. The results show that the Random Forest algorithm outperforms other models, achieving an accuracy of 89.5%, a precision of 97%, a recall of 83%, and an f1-score of 89%. This study demonstrates the effectiveness of the proposed preprocessing method and the potential of machine learning techniques in predicting diabetes may help clinicians identify individuals at risk of developing diabetes for early intervention.**

Keywords*— *Diabetes prediction, Machine Learning, K-Nearest Neighbors, SVM, XGBoost, Random Forest.

I. Introduction

Diabetes is a chronic (long-lasting) health condition that affects how your body turns food into energy. It is characterized by high levels of blood glucose (sugar) due to the body's inability to produce or effectively use insulin, a hormone that regulates blood sugar levels [1]. Therefore, there is a pressing need to explore new approaches for diabetes prevention, management, and treatment to reduce its burden on individuals, families, and healthcare systems.

Despite advancements in medical treatments and interventions, there is not a cure for diabetes yet, but early diagnosis and management of diabetes are crucial in preventing complications such as blindness, kidney failure, heart disease, and amputations [1]. There are treatments to reduce the chance of getting diabetes by losing weight, eating healthy food, and being active can really help.

The proposed study aims to contribute to existing studies in diabetes classification by comparing the performance of a few machine learning algorithms, namely K-Nearest Neighbors Algorithm (KNN), Support Vector Machine (SVM), eXtreme Gradient Boosting (XGBoost), and Random Forest. By using the Pima Indian Diabetes Database (PIDD) as the dataset, this study explored the effectiveness of these algorithms in predicting diabetes. This study discussed valuable insights into the potential of these algorithms for diabetes prediction and aid in identifying the best algorithm for diabetes classification. This study added to the knowledge of diabetes classification and contributed to the development of more accurate and effective predictive models for diabetes diagnosis.

Due to their excellent accuracy, adaptability, and robustness, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), XGBoost, and Random Forest are some of the most well-liked algorithms of this group. KNN is a non-parametric technique that may be applied to both regression and classification tasks. It doesn't make any assumptions about the distribution of the underlying data and is straightforward to comprehend and use. The new data point is then assigned to the class with the majority of the nearest neighbors after KNN determines the distances between it and the existing data points. KNN is frequently used in image recognition, recommendation systems, and medical diagnosis [13].

SVM is a potent method that may be applied to both regression and classification tasks. It is a binary classifier that uses a hyperplane to divide the input into two classes. By utilizing a kernel function, SVM can handle data that can be separated into linear and non-linear categories. SVM has been applied to a variety of tasks, including text classification, picture classification, and bioinformatics [15]. A tree-based ensemble technique that has gained popularity recently is called XGBoost. A high number of features may be handled by this scalable, accurate, and quick algorithm, which is also less prone to overfitting. By sequentially adding weak learners, XGBoost uses gradient boosting to enhance the performance of the model.

Another tree-based ensemble technique that is popular because of its excellent accuracy and capacity for handling complex data is Random Forest. To classify the data, it combines several decision trees and employs a voting system. Both categorical and continuous data may be handled by Random Forest, and it is resistant to outliers and missing values. Bioinformatics, image recognition, and credit risk analysis are just a few of the areas where Random Forest has been applied. Overall, the machine learning algorithms KNN, SVM, XGBoost, and Random Forest are well-liked and efficient for creating predictions and classifying data in a variety of domains. The type of data, the issue at hand, and the level of accuracy that is sought all play a role in choosing the best algorithm.

The objective of the paper about diabetes prediction was to develop a machine-learning model that can accurately predict the onset of diabetes in individuals based on their numbers of pregnancies, glucose, blood pressure, skin thickness, insulin, body mass index, diabetes pedigree function, and age. Many researchers are already conducting research on this topic; this research wants to develop machine learning that has the highest accuracy. The goal is to identify individuals who are at high risk of developing diabetes with high accuracy. Overall, the paper aims to provide a comprehensive and accurate diabetes prediction model that can help healthcare professionals identify high-risk individuals.

The proposed model for this experi”ent ’ncluded the K-Nearest Neighbors Algorithm (KNN), Support Vector Machine (SVM), and eXtreme Gradient Boosting (XGBoost). This study used these models for the reason that many researchers have used KNN, SVM, and RF as the model with varying outcomes [1][3][7][10]. This experiment uses these models to provide the maximum accuracy among them because few researchers use XGBoost.

This research used a PIDD dataset, the dataset consists of multiple independent variables that include the number of times a patient is pregnant, plasma glucose concentration, blood pressure, skin thickness, BMI, and so on. Whereas, there is only one dependent variable, which is the outcome consisting of the numbers 0 and 1. 1 signifies that the patient has diabetes and 0 does not. This dataset was originally gathered from the National Institute of Diabetes and Digestive and Kidney Diseases. By using K-Nearest Neighbor, Support Vector Machine, XGBOOST, and Random Forest Algorithm as the methods to classify the data, a comparison between those models is possible. The comparison will be based on multiple metrics, such as the model’s accuracy, F1 score, recall and precision [12].

Even though this dataset provides data from real patients in India, there are also a lot of outliers. Some columns have been detected to have the same exact value, which is dropped. There is also data that seems impossible to be true as this dataset is based on humans, such as a 0 BMI score, blood pressure of 0 mmHG, glucose concentration of 0, and skin thickness of 0mm. To overcome this issue, all null values are replaced by the average of each column; the average value is calculated by adding all the values divided by the number of rows that do not have a null value on that specific column. Last, the column insulin is dropped because 57% of it only contains null values, which will make the data unreliable.

II. Literature review

Previous studies focused on developing classification models for diabetes prediction using various machine learning algorithms. The first set of studies uses traditional algorithms like © Bayes, Random Forest, and KNN [1][4][3][8][9][10]. The Pima Indians Diabetes Database is commonly used as the dataset [1][4][6][7][8][9][10] One study using the DMP_MI algorithm outperformed other algorithms on accuracy and other classifier performance indicators, indicating potential for diabetes prediction [1]. A study using logistic regression analysis of PPG signal morphology also achieved high accuracy of 92.3% [2]. Another study of the KNN algorithm using 5000 samples generated by the criteria of the American diabetes association shows that fine KNN types have superior performance over coarse and cosine types, with a maximum accuracy rate of 90.36% [3].

The second set of studies uses more advanced algorithms such as Deep Learning, Fuzzy Logic based Diabetes Diagnosis System (FLDDS), and Enhanced and Adaptive-Genetic Algorithm-Multilayer Perceptron (EAGA-MLP) [5][6]. These studies have shown that deep learning and hybrid models can achieve high accuracy rates of up to 97.76%, with F-Score and precision values also exceeding 80% [6]. A study of SVM is shown to outperform decision trees and KNN with the highest accuracy of 90.23% using PIDD datasets [7].

The other studies compared multiple machine learning algorithms, including the Decision Tree, Gaussian NB, LDA, SVC, Extra Trees, AdaBoost, Perceptron, Logistic Regression, Gradient Boost Classifier, Bagging, and KNN. Resulting in logistic regression with the highest accuracy of 96% by using a diabetes dataset that contains 800 records and 10 attributes such as, Number of Pregnancies, Glucose Level, Blood Pressure, Skin Thickness, etc.[8]. One study of SVM-linear models provided the best accuracy of 0.89 and precision of 0.88 for the prediction of diabetes [10]. Overall, these studies have shown that various machine learning algorithms can be used to develop accurate and efficient classification models for diabetes prediction.

III. Methodology

In this study, the classification model that was used are K-Nearest Neighbors (KNN), XGBoost, Support Vector Machine (SVM),and Random Forest to tackle the problem at hand. Each model has distinctive qualities, and their combination enables us to experiment with various strategies in order to provide classification results that are accurate and trustworthy.

A. *Dataset Collecting*

The dataset that was used in the current study is well-known and commonly used in studies that are similar to it. It is known as the Pima Indians Diabetes Database and is widely used in studies on diabetes. The dataset includes a wide range of important medical factors, such as age, the number of pregnancies, glucose levels, and more. It also includes a target column called "outcome," which has binary values (0 for no diabetes, 1 for diabetes).

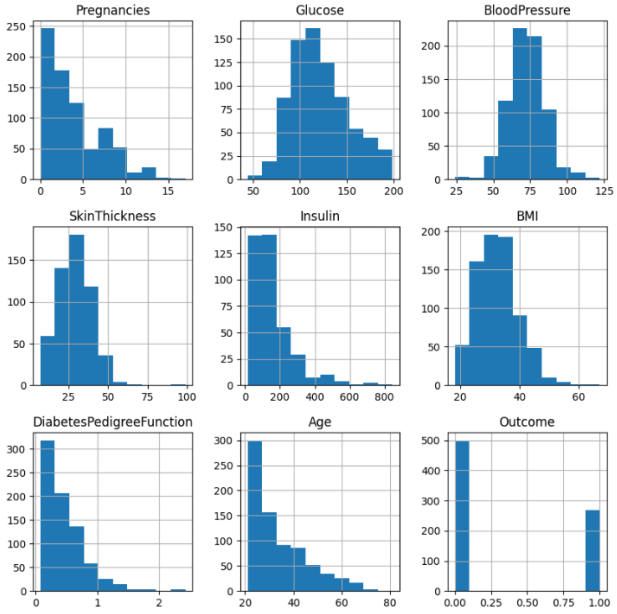

Fig. 1. Dataset Distribution.

Figure 1 illustrates the distribution of the dataset's key variables. The "Pregnancies" variable represents the number of times pregnant, ranging from 0 to 17, with the majority falling within the 0 to 5 range. The "Glucose" variable corresponds to plasma glucose concentration measured 2 hours after an oral glucose tolerance test, ranging from 50 to 200. The "Blood pressure" variable represents diastolic blood pressure in mmHg, with values ranging from 25 to 125 mmHg and a concentration around 75 mmHg. "Skin Thickness" refers to triceps skin fold thickness in mm. The "Insulin" variable represents 2-hour serum insulin levels in mu U/ml, with a concentration around 0 to 200. "BMI" stands for body mass index, primarily concentrated within the 30-40 BMI range. The "Diabetes pedigree function" ranges from 0.08 to 2.422, with a concentration around 0.08 to 0.78. Lastly, the "Age" variable spans from 20 to 80, with a majority falling within the 20-40 age range. The dataset comprises 500 non-diabetic results and 268 diabetic results.

B. *Data Pre-processing*

This dataset contains an imbalanced amount of output and zero values on the "output" column. Based on the amount of zeroes and ones that is portrayed using a countplot, the amount of zeroes clearly outweighs the amount of ones with almost twice its amount. To overcome this issue, the upscaling technique is used. Upscaling a dataset means increasing sample sizes to address imbalances. After using the upscaling method, the number of zeros and ones is finally balanced, with 500 zeros and 500 ones. The second issue with this dataset is impossible data—data that is impossible to obtain, knowing that it is collected from human beings. For example, a skin thickness of 0mm, a BMI score of 0, 0mm Hg blood pressure, 0 insulin level, and a plasma glucose concentration of 0 These data have one thing in common: all of them have a value of 0. Therefore, in order to resolve this problem, all zeros are replaced by null. then the null values are replaced by the mean of that column.

C. *Classification Model*

In this study, five different classification models was chosen for the analysis: K-Nearest Neighbors (KNN), XGBoost, Support Vector Machine (SVM), and Random Forest in order to capture diverse elements of the dataset and take advantage of the strengths of various techniques.

1) KNN

A data point is classified using KNN, a non-parametric classification technique, based on the consensus of its k nearest neighbors. Unlike other machine learning algorithms, KNN does not have an explicit training phase. During the training phase, the KNN algorithm simply stores its features and their corresponding labels. It calculates the distance between the new data and every other feature in the training set, the algorithm then chooses the K closest neighbors (the value K is adjusted manually), and then chooses the class label based on the majority vote of the neighbors. This study conducted experiments with various values of k and assessed the effectiveness of the model to find the ideal value. It's important to take into account how changing k will affect accuracy, precision, recall, and F1-score. In order to balance model accuracy and efficiency [13].

2) XGBoost

Extreme Gradient Boosting, or XGBoost, is a machine learning algorithm that is essentially a combination of multiple weaker prediction models, typically decision trees, that is combined with the principles of gradient boosting. XGBoost works by iteratively improving the overall model by minimizing each loss during each iteration through gradient descent. The training phase consists of model creation and error calculation for each iteration. It then continues fitting a new model while updating the overall model by adjusting its weights to maximize the accuracy of the overall model. With all of the models combined, it is then used to make the final prediction.This study undertook a thorough tuning process for the hyperparameters in order to enhance the performance of XGBoost. This study tested with different hyperparameter settings, including learning rate, maximum tree depth, and number of estimators. This study discovered the collection of hyperparameters through this method that produced the most accurate categorization outcomes [14].

3) Support Vector Machine (SVM)

Support Vector Forest, or SVM, is a machine learning model that separates the target values of the data in such a way that all of the target values are separated by one or more hyperplanes. The goal during the training phase is to reach the maximum distance between the closest training data and the hyperplane in order to achieve the best separator between classes. Although, in some cases, the data is not linearly separable. In this case, the kernel trick is put to use. The kernel trick transforms the data into a higher-

dimensional space, where the data is then separable. However, using the kernel trick, the hyperplane is not separated linearly. To determine the ideal SVM configuration, this study tested a variety of hyperparameters, including the regularization parameter (C) and the kernel coefficient (gamma). This study sought to determine the ideal kernel function and hyperparameter combination for our classification problem by comparing the performance of various SVM variations [15].

4) Random Forest

Random Forest is a machine learning algorithm that uses a combination of multiple decision trees to make predictions. The random forest algorithm starts off by creating a bunch of decision trees. Each decision tree is then assigned different random subsets of features based on the given data. They will now start to train themselves to make predictions based on the given features. The tree keeps on making subsets until it reaches the correct prediction for each class or target. The final predictions are made by these trees, and each tree will make a prediction. Usually, not all trees have the same prediction; therefore, the final prediction is made by choosing the majority of the prediction made.This study sought to find the sweet spot that offered a fair compromise between model complexity and classification performance by examining the accuracy, precision, recall, and F1-score for each number of trees. To learn more about the most important features for categorization, this study also looked at the feature importance metrics offered by Random Forest [16].

D. Evaluation Metrics

This study sought to gain a thorough grasp of the advantages and disadvantages of each categorization model by applying these evaluation measures jointly. By comparing and contrasting the models according to their recall, accuracy, precision, and F1-score, this study was able to determine which models would be most effective for our particular categorization task.

1) Accuracy

A frequent evaluation criterion used to assess the effectiveness of a classification model is accuracy. It displays the percentage of examples (or predictions) that were successfully categorized out of all the cases in a dataset. Accuracy is calculated as follows True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [12].

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

The accuracy metric gave a broad indication of how successfully the models identified instances in the dataset. It quantified the percentage of occurrences that were correctly tagged and served as a key performance indicator for the model.

2) Precision

Precision is an evaluation metric that assesses the ability of a classification model to correctly identify positive instances (true positives) among all instances predicted as positive. Out of all instances projected as positive, it quantifies the percentage of correctly predicted positive instances. Precision is calculated as follows True Positive (TP) and False Positive (FP) [12].

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

Precision gave insights into the model's capacity to avoid false positives and accurately categorize positive occurrences by measuring the ratio of true positives to the sum of true positives and erroneous positives.

3) Recall

Recall, also referred to as sensitivity or the percentage of real positive occurrences in a dataset, is an evaluation metric that measures a classification model's accuracy in identifying positive cases. It quantifies the proportion of correctly predicted positive instances out of the total number of actual positive instances. Recall is calculated as follows True Positive (TP) and False Negative (FN) [12].

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

The model's sensitivity in correctly recognizing positive cases without missing any was demonstrated by computing the ratio of true positives to the total of true positives and false negatives. The model was able to successfully detect the majority of positive events, as evidenced by a greater recall value.

4) F1-Score

The F1-score is a metric that gives a fair assessment of the effectiveness of a classification model by combining precision and recall into a single value. It considers both precision (ability to identify positive occurrences with accuracy) and recall (ability to record all actual positive examples). F1-score is calculated as follows True Positive (TP), False Positive (FP) and False Negative (FN) [12].

$$F1 = \frac{2 * TP}{(2 * TP + FP + FN)} \quad (4)$$

This metric simultaneously used both measures to calculate the harmonic mean of precision and recall. This study was able to evaluate the model's performance in terms of recall and precision thanks to the F1-score, which also gave us a general idea of the model's classification abilities.

E. Machine Learning System

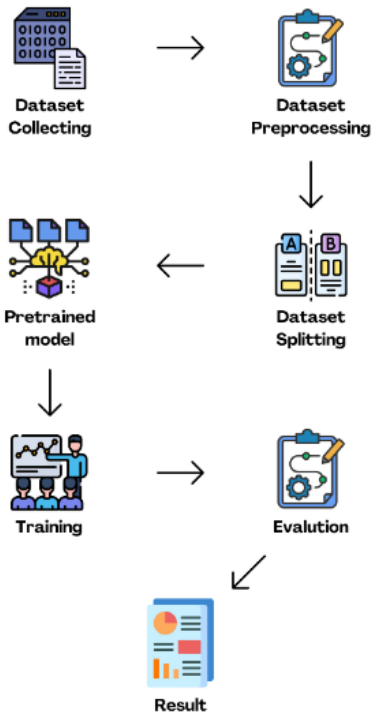

Fig. 2. The workflow of the whole process.

The proposed machine learning system is shown in Figure 2. We used PIDD dataset for the model trained using this data in order to make predictions. In Data Pre-processing, . After the Data Pre-processing, we split the data into test 20% and train 80% data. We made use of K-Nearest Neighbors (KNN), XGBoost, Support Vector Machine (SVM), and Random Forest models to classify the diabetes dataset. After that, the model algorithm builds a prediction model during the training phase by learning from the supplied training data. To determine the model's accuracy and generalizability after training, performance must be examined. We used evaluation metrics such as accuracy, precision, recall, and F1 score to measure the model's effectiveness in making predictions or decisions[11].

F. Results and Discussion

In this study, we evaluated the performance of four different machine learning models: K-Nearest Neighbors (KNN), XGBoost, Random Forest, and Support Vector Machine (SVM). The models were implemented using standard settings, with KNN set to 3 neighbors.

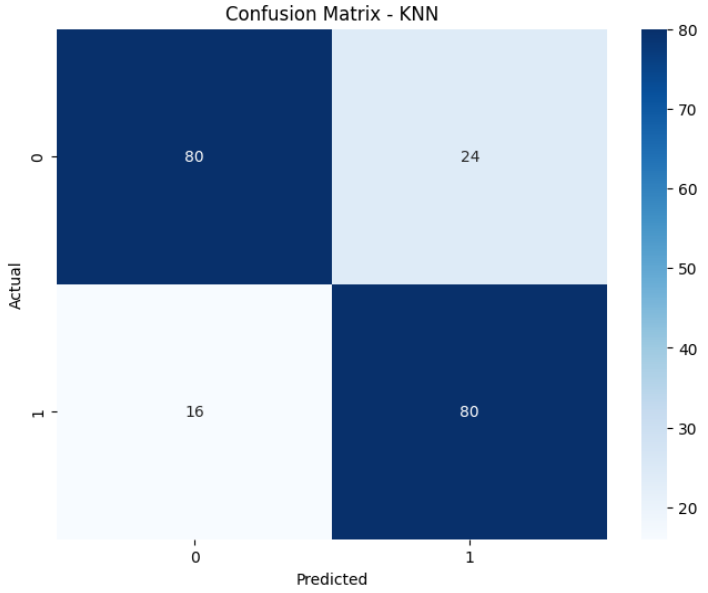

Fig. 3. Confusion Matrix of KNN.

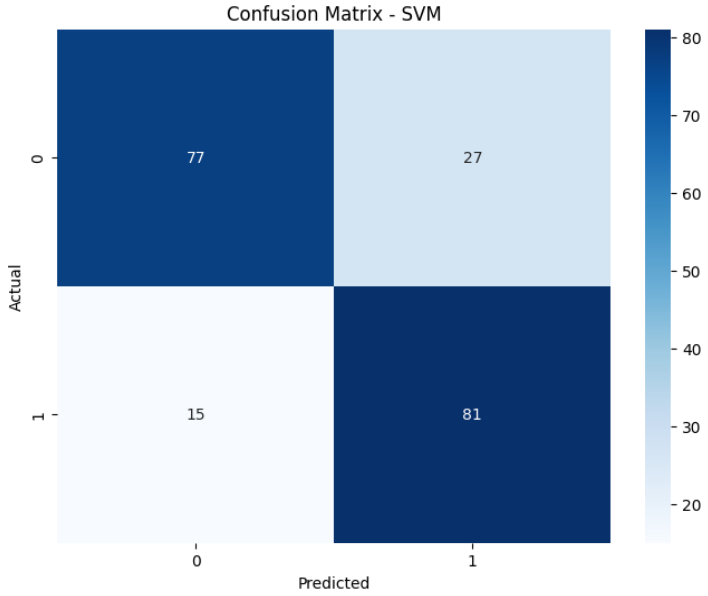

Fig. 4. Confusion Matrix of SVM

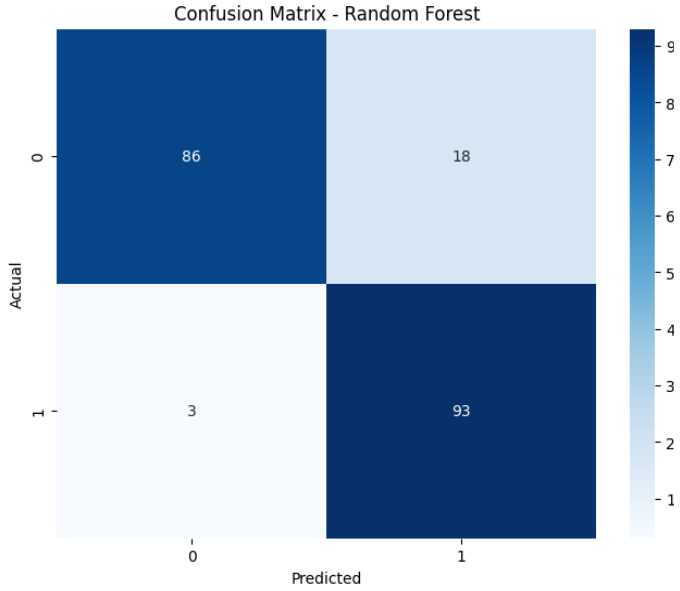

Fig. 5. Confusion Matrix of Random Forest.

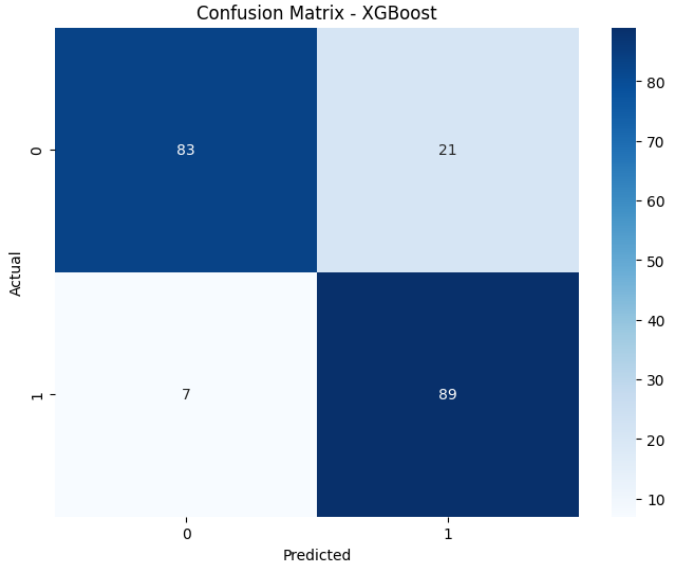

Fig. 6. Confusion Matrix of XGBoost.

Figures 3 to 6 present the confusion matrices of the classification models. These matrices provide a detailed overview of the model's performance in terms of correct and incorrect predictions. In a confusion matrix, the top left quadrant represents true positive predictions, while the bottom right quadrant represents true negative predictions. The figures reveal that all the models display a relatively low count of false negatives, indicating their ability to accurately identify positive instances. This implies that the models are successful in capturing the majority of actual positive cases. The low false negative count across the models indicates their effectiveness in avoiding the omission of positive instances, which is crucial in tasks where the consequences of missing positive cases are significant.

Model	Accuracy	Precision	Recall	F1
KNN	0.8	0.8	0.8	0.8
XGBoost	0.86	0.87	0.86	0.86
Random Forest	0.895	0.9	0.9	0.89
SVM	0.79	0.79	0.79	0.79

Table 1. Comparative Results of the Classification Models.

The KNN model achieved an accuracy of 0.8, precision of 0.8, recall of 0.8, and an F1-score of 0.8. These results indicate that the KNN model performs reasonably well in terms of overall accuracy, but it has a relatively lower recall compared to precision. This suggests that while the model correctly identifies a high proportion of positive instances (precision), it may miss some true positive instances (recall).

On the other hand, the XGBoost model demonstrated superior performance with an accuracy of 0.86, precision of 0.87, recall of 0.86, and an F1-score of 0.86. These results indicate that the XGBoost model outperforms the KNN model in terms of accuracy, precision, recall, and F1-score. It shows a better balance between precision and recall, suggesting that the XGBoost model has a higher capability of correctly classifying both positive and negative instances.

The Random Forest model achieved the highest accuracy among all the models with a value of 0.895. It also demonstrated excellent precision of 0.9, recall of 0.9, and an F1-score of 0.89. These results indicate that the Random Forest model performs remarkably well in terms of accuracy, precision, recall, and F1-score. It shows a high precision, meaning that it has a low false positive rate, and a reasonable recall, implying that it effectively captures a significant proportion of true positive instances.

The SVM model achieved an accuracy of 0.79, precision of 0.79, recall of 0.79, and an F1-score of 0.79. These results indicate that the SVM model has a relatively lower accuracy compared to the other models. Additionally, it exhibits a similar trend to the KNN model, with a higher precision than recall. This suggests that the SVM model may misclassify some positive instances, leading to a lower recall rate.

Future research directions include exploring feature engineering techniques, optimizing hyperparameters, investigating ensemble methods, implementing rigorous cross-validation, addressing imbalanced data, enhancing interpretability, conducting external validation, and performing comparative studies. These avenues of study aim to improve the models' performance, robustness, and applicability in real-world scenarios.

G. Conclusion

In conclusion, the evaluation of multiple classification models revealed varying levels of performance. The XGBoost and Random Forest models demonstrated superior overall performance, outperforming the KNN and SVM models in terms of accuracy, precision, recall, and

F1-score. The XGBoost model exhibited the highest accuracy, precision, recall, and F1-score, showcasing its effectiveness in correctly classifying both positive and negative instances. Similarly, the Random Forest model achieved the highest accuracy and demonstrated excellent precision, recall, and F1-score, effectively capturing true positive instances while maintaining a low false positive rate. The KNN and SVM models performed reasonably well in terms of accuracy, but they showed a relatively lower recall compared to precision, suggesting potential missed true positive instances.

However, further analysis and comparison with other models are necessary to fully assess its effectiveness. Overall, the XGBoost and Random Forest models emerged as the top performers, highlighting their strong predictive capabilities for similar classification tasks, while the KNN and SVM models may benefit from further optimization.

REFERENCES

[1] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng, and D. N. Davis, "DMP_MI: An Effective Diabetes Mellitus Classification Algorithm on Imbalanced Data With Missing Values," in IEEE Access, vol. 7, pp. 89980-89988, July 2019. doi: 10.1109/ACCESS.2019.2924394.

[2] Y. K. Qawqzeh, A. S. Bajahzar, M. Jemmali, M. M. Otoom, and A. Thaljaoui, "Classification of Diabetes Using Photoplethysmogram (PPG) Waveform Analysis: Logistic Regression Modeling," BioMed Research International, vol. 2020, Article ID 3764653, 10 pages, 2020. doi: 10.1155/2021/3764653.

[3] A. Ali, M. Alrubei, L. F. Mohammed Hassan, M. Al-Ja'afari, and S. Abdulwahed, "Diabetes Classification Based on KNN," International Journal of Advanced Computer Science and Applications, vol. 10, no. 7, pp. 175-180, 2019. doi: 10.14569/IJACSA.2019.0100722.

[4] S. Deepti and S. S. Dilip, "Prediction of Diabetes using Classification Algorithms," in Proceedings of the 2nd International Conference on Inventive Systems and Control, Coimbatore, India, January 2018, pp. 674-678. doi: 10.1109/ICISC.2018.8399441.

[5] K. Santosh, B. Bharat, S. Debabrata, and K. C. Dilip, "Classification of Diabetes using Deep Learning," in Proceedings of the International Conference on Communication and Signal Processing, India, July 2020, pp. 645-649. doi: 10.1109/ICCSP48502.2020.9181027.

[6] M. Sushruta, K. T. Hrudaya, K. M. Pradeep, K. B. Akash, and P. Paolo, "EAGA-MLP—An Enhanced and Adaptive Hybrid Classification Model for Diabetes Diagnosis," Sensors, vol. 20, no. 14, Article ID 4036, 2020. doi: 10.3390/s20144036.

[7] S. H. Abdulhakim, I. Malaserene, and L. A. Anny, "Diabetes Mellitus Prediction using Classification Techniques," International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2, pp. 381-387, 2020. doi: 10.35940/ijitee.L8415.129220.

[8] A. Mujumdar and V. V. Dr., "Diabetes Prediction using Machine Learning Algorithms," Procedia Computer Science, vol. 170, pp. 1134-1139, 2020. doi: 10.1016/j.procs.2020.03.164.

[9] S. Saru and S. Subashree, "Analysis and Prediction of Diabetes Using Machine Learning" (2019). International Journal of Emerging Technology and Innovative Engineering, Volume 5, Issue 4, April 2019. https://ssrn.com/abstract=3368308

[10] H. Kaur and V. Kumari , "Predictive modeling and analytics for diabetes using a machine learning approach". Applied Computing and Informatics, Vol. 18 No. 1/2. 90-100, 2022. https://doi.org/10.1016/j.aci.2018.12.004

[11] R. Saxena, S. K. Sharma, M. Gupta, G. C. Sampada, "A Novel Approach for Feature Selection and Classification of Diabetes Mellitus: Machine Learning Methods", Computational Intelligence and Neuroscience, vol. 2022, Article ID 3820360, 11 pages, 2022. https://doi.org/10.1155/2022/3820360

[12] Lever, J., Krzywinski, M. & Altman, N. "Classification evaluation". Nat Methods 13, 603–604 (2016). https://doi.org/10.1038/nmeth.3945

[13] W. Xing and Y. Bei, "Medical Health Big Data Classification Based on KNN Classification Algorithm," in IEEE Access, vol. 8, pp. 28808-28819, 2020, doi: 10.1109/ACCESS.2019.2955754.

[14] [M. Ma et al., "XGBoost-based method for flash flood risk assessment," Journal of Hydrology, vol. 598, p. 126382, Jul. 2021, doi: https://doi.org/10.1016/j.jhydrol.2021.126382.

[15] D. A. Otchere, T. O. Arbi Ganat, R. Gholami, and S. Ridha, "Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models," Journal of Petroleum Science and Engineering, vol. 200, p. 108182, May 2021, doi: https://doi.org/10.1016/j.petrol.2020.108182.

[16] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi and S. Homayouni, "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 6308-6325, 2020, doi: 10.1109/JSTARS.2020.3026724.