COMPARISON OF RANDOM FOREST ALGORITHM ACCURACY WITH XGBOOST USING HYPERPARAMETERS

[1]Kevin Stefanus, [2]Hironimus Leong
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]marlon.leong@unika.ac.id

ABSTRACT

Diabetes is one of the most dangerous diseases in the world and many people do not realizethat they have diabetes in them. So many factors affect the occurrence of diabetes such as pregnancies, glucose, blood pressure, skinthickness, insulin, BMI, diabetes pedigree function, and age. so diabetes threatens silently and will appear suddenly. Therefore, this study will make a diabetes prediction using Random Forest and XGBoost algorithms. The model will be evaluated with accuracy, F1-Score, recall, and precision. for randomization or random s

tate will use random states 0 and 45. The results obtained from the comparison of these two algorithms are the highest accuracy of the random forest algorithm has a value of 88,98% while the highest accuracy of XGBoost gets an accuracy value of 87,00% at random state 45 and data division 90/10, while random state 0 random forest has the highest accuracy value also with a value of 78,43% with data division 90/10 while XGBoost gets the highest accuracy value of 76,47% at data division 90/10. It can be concluded that random forest is better at predicting diabetes data than the XGBoost algorithm.

Keywords: Prediction, Random Forest, XGBoost, Accuracy, Hyperparameters

1. INTRODUCTION

Diabetes is a life-threatening disease that often goes unnoticed and is influenced by various factors including pregnancy, glucose levels, blood pressure, skin thickness, insulin levels, BMI, diabetes pedigree function, and age. In the field of computer science, there are several algorithms available to process these diabetes-related factors and predict the occurrence of diabetes.

This research aims to utilize two algorithms, namely Random Forest and XGBoost with hyperparameters, to predict diabetes early. A comparison will be made between the two algorithms using hyperparameters to determine which algorithm has higher accuracy in predicting diabetes. The dataset used for this research consists of 768 data obtained from the Pima Indians Diabetes Database (PIDD).

The performance of both algorithms is evaluated based on various criteria such as Precision, F-1 Score, and Recall.

2. LITERATURE STUDY

Nasution et al (2021). Comparison of Accuracy of NaÏve Bayes Algorithm and Xgboost Algorithm in Diabetes Disease Classification [1]. This journal, discusses the comparison of the classification performance of the Supervised Learning Algorithm, namely Naïve Bayes and

XGBoost, and handling missing values on the dataset, and discusses the Grid Search method as an optimization based on classification accuracy performance using the Pima Indians Diabetes Database dataset.

GivarGivari et al (2022). Comparison of SVM, Random Forest, and XGBoost Algorithms for Determining Credit Application Approval [2]. This journal discusses the comparison of three algorithms, namely SVM, random forest, and XGBoost to determine approval and eligibility in providing credit to individuals. The results of research using SVM, random forest, and XGBoost algorithms get the highest accuracy, recall, and precision values in the XGBoost model.

Mursianto et al (2021). Comparison of Random Forest and XGBoost Classification Methods and Implementation of SMOTE Technique in Rain Prediction Cases [3]. This journal discusses how to classify the prediction of rain predictions on the following days, using several classification methods, namely Random Forest, XGBoost, and XGBoost. The weather prediction system that we have made gets the highest level of accuracy obtained by Random Forest classification.

Supriyadi et al (2020). Application of Random Forest Algorithm to Determine the Quality of Red Wine [4]. In this study, it discusses classifying red wine. In this study it was carried out by applying machine learning by comparing three data mining algorithms, namely, Decision Tree, Random Forest, and Support Vector Machine (SVM), from the results of research that have been done by comparing the three algorithms, Random Forest produces the best accuracy among other algorithms that have been tested.

Hendrawan, I. R. (2022). COMPARISON OF NAÏVE BAYES, SVM AND XGBOOST ALGORITHMS IN TEXT CLASSIFICATION OF COMMUNITY SENTIMENT TOWARDS LOCAL PRODUCTS IN INDONESIA [5]. In this study, it discusses analyzing each customer review of a local item using the naive Bayes, XGBoost, and SVM algorithms. There are 6 training model schemes namely TF-IDF + NaïveBayes, Word2vec + NaïveBayes, TFIDF + SVM, Word2vec + SVM, TF-IDF + XGBoost and Word2vec + XGBoost. Based on the research that has been done, the Word2vec + XGBoost combination produces a higher F1 Score of 0.941 followed by TF-IDF + XGBoost 0.940.

Syukron et al (2020). COMPARISON OF SMOTE RANDOM FOREST AND SMOTE XGBOOST METHODS FOR HEPATITIS C DISEASE LEVEL CLASSIFICATION ON IMBALANCE CLASS DATA [6]. This study looks for the best SMOTE algorithm between random forest and xgboost in predicting hepatitis c with imbalance class data. the results of the study show that the random forest algorithm and XGBoost have 75% accuracy but the recall value is less than 2%.

Andryan et al (2022). PERFORMANCE COMPARISON OF XGBOOST ALGORITHM AND SUPPORT VECTOR MACHINE (SVM) ALGORITHM FOR BREAST CANCER DIAGNOSIS [7]. In this study, researchers wanted to compare the performance of the XGBoost algorithm with the SVM algorithm for diagnosing breast cancer. The method used in this research is Knowledge Data Discovery (KDD) using the XGBoost and SVM algorithms, then classification is carried out to determine whether the cancer analyzed is benign or malignant. The performance results obtained after conducting research using both algorithms are Xgboost which has the best accuracy and ROC AUC.

Derisma, D. (2020). Comparison of Algorithm Performance for Heart Disease Prediction with Data Mining Techniques [8]. The methodology of this research is to collect datasets then conduct a literature study and then select a model using the naive Bayes algorithm, random forest, and neural network. After that, researchers conducted training on the dataset and then evaluated predictions with AUC, CA, F1, Precision, Recall, Confusion Matrix, and ROC analysis.

Erdiansyah et al (2022). Comparison of K-Nearest Neighbor and Random Forest Methods in Predicting the Accuracy of Classification of Wart Disease Treatment [9]. This research focuses on comparing the K-Nearest Neighbor classification method with Random Forest to see the level of accuracy in predicting the success of wart disease treatment. Based on the results of testing the K-Nearest Neighbor and Random Forest methods, K-Nearest has higher accuracy.

Apriliah et al (2021). Prediction of the Likelihood of Diabetes in the Early Stage Using the Random Forest Classification Algorithm [10]. The research contains a comparison of Support Vector Machine, Naive Bayes, and Random Forest to detect diabetes early. The methodology used is to do data mining which is then carried out data preprocessing, after which the data is processed in the algorithm to be compared, the performance of the three algorithms is evaluated on various measures such as Precision, Accuracy, F-Measure, and Recall.

3. RESEARCH METHODOLOGY

3.1 Research Reference Exploration

Searched 10 relevant research journals focusing on random forest algorithms and XGBoost. These journals will serve as valuable references for this research project. Obtaining sufficient references will provide a comprehensive understanding of the random forest algorithm and XGBoost, thus facilitating the research process.

3.2 Looking for data to be processed

This study incorporates the dataset known as the Pima Indians Diabetes Database (PIDD), which can be accessed from the website https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database. The dataset consists of 768 rows and 9 columns, offering a comprehensive collection of information related to diabetes.

3.3 Pre-processing Data

After obtaining the data, it is examined to identify any null values present in the attributes such as Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI. Subsequently, any identified null values in these attributes are replaced with the median value of the respective attribute [1].

Subsequently, the data is assessed for balance by comparing the occurrences of values 0 and 1 in the target column. If the data is found to be imbalanced, it is necessary to balance it by oversampling the minority class, which in this case is the value 1. Oversampling involves replicating instances of the minority class to rectify the imbalance. This technique offers the advantage of increasing the representation of the minority class and reducing errors in the dominant class [2].

Once oversampling is performed, it is essential to examine the data for outliers. The outliers within the dataset are then eliminated using the IQR (Interquartile Range) technique. This method involves utilizing the IQR to establish the lower and upper limits. Any values that fall outside these limits are considered outliers and can be removed from the dataset [3].

3.4 Fit the Train and Test Data

The dataset needs to be divided into training data and testing data. In this scenario, the data will be split into various ratios: 90% for training and 10% for testing, 80% for training and 20% for testing, 70% for training and 30% for testing, 60% for training, and 40% for testing.

3.5 Hyperparameters Tuning

Before entering into the algorithm, the model will enter the selection of hyperparameters that are suitable for the model. between random forest and XGboost have different hyperparameter settings needed. Grid search will determine the hyperparameters that are suitable for the model.

3.6 Implementing data into algorithms

For the first experiment, a random state value of 0 will be used. The implementation results will include model evaluation metrics such as accuracy, precision, recall, and F1-score.

The process will iterate through all the split test-train data and provide the respective results. Following that, the random state will be changed to 45, and the process will repeat similarly to the previous experiment. The program will run until all the split test-train data has been fitted and evaluated to obtain the results.

3.7 Model Evaluation

Utilizing Recall, Precision, F-1 Score, and Confusion Matrix, the model was assessed. Precision measures the percentage of instances or samples that are accurately categorized among those that are classed as positive. recall is one of the evaluation metrics used to measure the extent to which the model can correctly identify and recall positive classes. More specifically, recall measures how many of the total positive class instances were successfully found by the model. A binary classification method that divides information into "positive" and "negative" is evaluated using the F1-Score.

The confusion matrix is a visual representation of the algorithm's performance, displaying the tabulation of observed and predicted classes along with associated statistics. It is used to evaluate the performance of a method by calculating measures such as sensitivity, specificity, precision, and accuracy.

4. RESULT AND DISCUSSION

The results obtained after conducting several tests for data divided into 90%/10%, 80%/20%,	**Evaluation Result**	**Train / Test Random Forest = 0**

70%/30% and 60%/40% with random state 0 and 45. for random state = 45 will be tested 5 times and the final results in the form of an average value of the test results.**Algorithm**					
		90/10	**80/20**	**70/30**	**60/40**
Random Forest	Accuracy	78,43%	79,20%	78,14%	79,20%
	Precision	76,66%	75,00%	76,19%	75,89%
	Recall	85,18%	88,23%	83,11%	85,00%
	F1-Score	80,70%	81,08%	79,50%	80,18%
Xgboost	Accuracy	76,47%	74,25%	70,86%	76,23%
	Precision	74,19%	72,72%	70,37%	73,63%
	Recall	85.18%	78,43%	74,02%	81,00%
	F1-Score	79,31%	75,47%	72,15%	77,14%

Table 4.1 Table Result with random state = 0

These results show that Random Forest generally performs better than XGBoost in terms of accuracy, precision, recall, and F1-score at different training and testing ratios. The highest accuracy is owned by the 80/20 data split with 79.20% accuracy and 60/40 with 79.20% accuracy. although with the same accuracy results, the precision, recall and F1-Score values on the 80/20 split are greater than with the 60/40 split. The division of training data and test data affects accuracy because accuracy is calculated based on the comparison between predictions made by the model and actual values in the test data. In the case of data division 80/20 and 60/40 the resulting accuracy is the same, this is because the dataset used is relatively small, the division of training data and test data can cause the same or very similar accuracy between training data and test data. Although sometimes the same or similar accuracy between training and test data can occur, it should be noted that other evaluation metrics such as precision, recall, and F1 score can differ between the two datasets.

Algorithm	**Evaluation Result**	**Hyperparameters Random Forest = 0**			
		90/10	**80/20**	**70/30**	**60/40**

Random Forest	n_estimators	50	100	50	100
	max_features	sqrt	sqrt	sqrt	sqrt
	max_depth	10	None	None	None
	min_samples_split	1	5	5	5
	min_samples_leaf	2	1	2	1
	criterion	gini	entropy	gini	entropy
	bootstrap	TRUE	TRUE	FALSE	TRUE
Xgboost	n_estimators	500	50	50	100
	learning_rate	0.1	0.1	0.1	0.1
	max_depth	10	10	5	5
	min_child_weight	5	5	5	5
	subsample	0.9	0.8	0.9	0.9
	colsample_bytree	0.9	1.0	0.8	0.8
	gamma	1	1	2	1

Table 4.2 Table Hyperparameters with random state = 0

The following is the result of the best hyperparameters obtained from the GridSearch function where each hyperparameters set will be tried one by one into the data so as to find the most optimal hyperparameters.

In the table, it can be seen that random forest hyperparameters for n-estimators use 50 and 100, for max_features using SQRT, max_depth on average uses none, min_sample_Split on average uses a value of 5, min_saple_leaf on divisions 90/10 and 70/30 uses a value of 2 while for divisions 80/20 and 60/40 uses a value of 1, criterion on divisions 90/10 and 70/30 uses the type gini while for divisions 80/20 and 60/40 uses the type entropy, and for boostrap divisions 90/10, 80/20 and 60/40 boostrap = true, while for division 70/30 boostrap = false.

The results for testing with random state 45 were tested up to 5 times, this was done to get the average value of accuracy, precision, recall, and F1-score.

Algorithm	**Evaluation Result**	**Average Results**			
		90/10	**80/20**	**70/30**	**60/40**
	Accuracy	88,98%	84,32%	82,27%	83,49%

Random Forest	Precision	89,33%	81,25%	78,70%	79,98%
	Recall	90,90%	89,82%	86,19%	88,79%
	F1-Score	90,10%	85,32%	82,27%	84,15%
Xgboost	Accuracy	87,00%	82,37%	81,25%	82,29%
	Precision	84,92%	78,36%	76,98%	77,54%
	Recall	89,52%	87,72%	86,66%	89,31%
	F1-Score	87,54%	82,77%	81,53%	83,28%

Table 4.3 Tabel of Average Test 5 with Random State = 45

The table above is the result of the average accuracy, precision, recall, and F1-Score values of the Random Forest and XGBoost algorithms using Hyperparameter with Random State 45. From the data, we can see if the accuracy, precision, recall and F1-Score results of the Random Forest algorithm are greater than the XGBoost algorithm even though the distance between the values is small.

The greatest accuracy, precision, recall, and F1-Score values of the Random Forest algorithm are in the 90/10 data division with an accuracy of 88.98%, precision of 89.33%, recall of 90.90%, and F1-Score of 90.10%. Meanwhile, the greatest accuracy, precision, recall, and F1-Score values of the XGBoost algorithm are in the 90/10 data division with an accuracy of 87.00%, precision of 84.92%, recall of 89.52% and F1-Score of 87.54%.

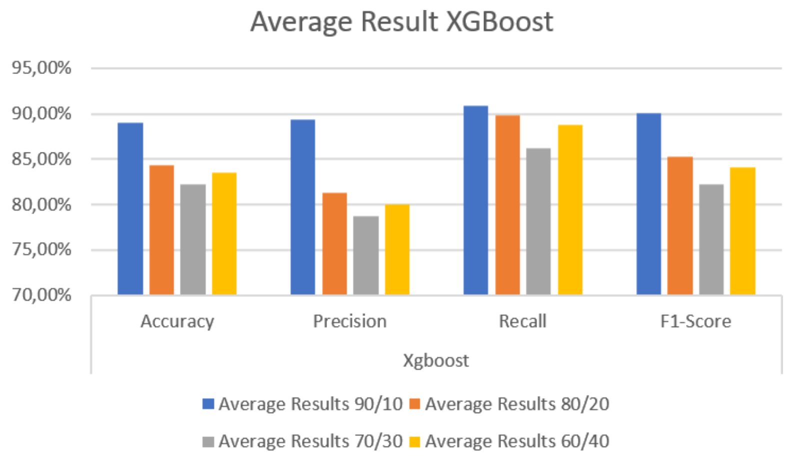

Figure 4.3 Diagram of Average result Random Forest with Random State = 45

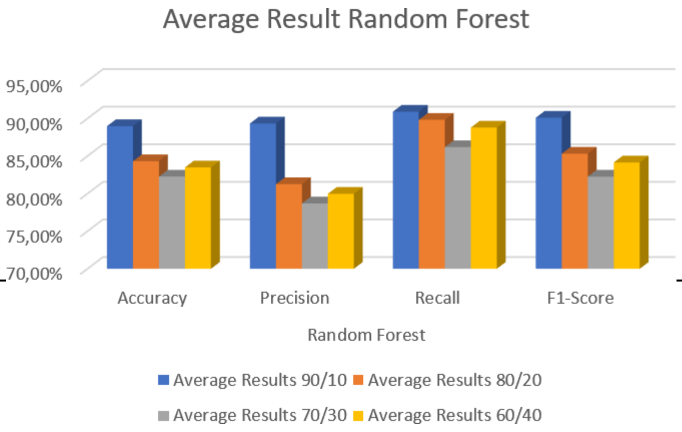

Figure 4.4 Diagram of Average result XGBoost with Random State = 45

5. CONCLUSION

This research concludes that the algorithm that has the highest accuracy and the best model evaluation in managing diabetes data is the Random forest algorithm with the highest accuracy of 88.98%, precision of 89.33%, recall of 90.90%, and F1-Score of 90.10% using random state 45.

In experiments using random state 0 the Random Forest algorithm also gets higher accuracy, precision, recall, and F1-Score scores than XGBoost.

Therefore in this study, the Random Forest algorithm is better at finding accuracy, precision, recall, and F1-Score for PIMA Indian Diabetes data. The use of hyperparameters greatly affects the results obtained from the model.

For future research, it can be suggested to compare the accuracy of random forest and XGboost algorithms without balancing data and using other hyperparameter combinations

DAFTAR PUSTAKA

[1] M. K. Nasution, RD. R. Saedudin, V. P. Widartha “perbandingan akurasi algoritma naïve Andryan, M. R., Fajri, M., & Sulistyowati, N. (2022). KOMPARASI KINERJA ALGORITMA XGBOOST DAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) UNTUK DIAGNOSA PENYAKIT KANKER PAYUDARA. *Jurnal Informatika dan Komputer*, 1-5. https://ejournal.akakom.ac.id/index.php/jiko/article/download/500/pdf

[2] Apriliah, W., Kurniawan, I., Baydhowi, M., & Haryati, T. (2021, januari). Prediksi Kemungkinan Diabetes Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest. *Jurnal Sistem Informasi, 10*, 163-171. http://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/1129

[3] Derisma. (2020). Perbandingan Kinerja Algoritma Untuk Prediksi Penyakit Jantung Dengan Teknik Data Mining. *Journal of Applied Informatics and Computing, 4*, 84-88. https://www.researchgate.net/publication/344979585_Perbandingan_Kinerja_Algoritma_untuk_Prediksi_Penyakit_Jantung_dengan_Teknik_Data_Mining

[4] Erdiansyah, U., Lubis, A. I., & Erwansyah, K. (2022, Januari). Komparasi Metode K-Nearest Neigbhor dan Random Forest Dalam Prediksi Akurasi Klasifikasi Pengobatan Penyakit Kutil. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 208-214. https://ejurnal.stmik-budidarma.ac.id/index.php/mib/article/view/3373

[5] Givari, M. R., Sulaeman, M. R., & Umaidah, Y. (2022). Perbandingan Algoritma SVM, Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit. *JURNAL NUANSA INFORMATIKA* , 1-9. https://journal.uniku.ac.id/index.php/ilkom/article/view/5406/2901

[6] Hendrawan, I. R. (2022). PERBANDINGAN ALGORITMA NAÏVE BAYES, SVM DAN XGBoost Dalam Klasifikasi Teks Sentimen Masyarakat Terhadap Produk Lokal Di Indonesia . *Jurnal TRANSFORMASI* , 1-6. https://ejournal.stmikbinapatria.ac.id/index.php/JT/article/view/295/191

[7] Mursianto, G. A., Falih, I. M., Irfan, M., Sakinah, T., & Prasvita, D. S. (2021). Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta. *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, 1-10. https://conference.upnvj.ac.id/index.php/senamika/article/download/1627/1340

[8] Nasution, M. K., Saedudin, R. R., & Widartha, V. P. (2021). PERBANDINGAN AKURASI ALGORITMA NAÏVE BAYES DAN ALGORITMA XGBOOST PADA KLASIFIKASI PENYAKIT DIABETES. *e-Proceeding of Engineering* , 1-8. https://docplayer.info/222292164-Perbandingan-akurasi-algoritma-naive-bayes-dan-algoritma-xgboost-pada-klasifikasi-penyakit-diabetes.html

[9] Supriyadi, R., Gata, W., Maulidah, N., & Fauzi, A. (2020). Penerapan Algoritma Random Forest Untuk Menentukan . *JURNAL ILMIAH EKONOMI DAN BISNIS*, 1-9. https://journal.stekom.ac.id/index.php/Bisnis/article/download/247/182

[10] Syukron, M., Santoso, R., & Widiharih, T. (2020). PERBANDINGAN METODE SMOTE RANDOM FOREST DAN SMOTE XGBOOST. *JURNAL GAUSSIAN, 9*, 227 - 236. https://ejournal3.undip.ac.id/index.php/gaussian/article/download/28915/24507