

# PERBANDINGAN AKURASI ALGORITMA ADABOOST DAN ALGORITMA LIGHTGBM UNTUK KLASIFIKASI PENYAKIT DIABETES

Rangga Ahsana<sup>1</sup>, Rd. Rohmat Saedudin<sup>2</sup>, Vandha Pradwiyasma Widartha<sup>3</sup>

<sup>1, 2, 3</sup> Universitas Telkom, Bandung

ranggaahsana@student.telkomuniversity.ac.id<sup>1</sup>, rdrohmat@telkomuniversity.ac.id<sup>2</sup>, vandhapw@telkomuniversity.ac.id<sup>3</sup>

---

## Abstrak

Diabetes dikenal sebagai penyakit epidemi yang berdampak di hampir setiap negara, kelompok usia, dan faktor ekonomi di seluruh dunia. Di Indonesia sendiri, berdasarkan data dari IDF tahun 2020, dari total populasi orang dewasa sebanyak 172 juta jiwa, 10,5 juta jiwa mengidap diabetes. Ketika merancang *software* atau perangkat lunak diagnosis medis, prediksi penyakit menjadi salah satu tugas yang memakan banyak waktu dikarenakan terdapat banyak aspek yang masing-masing harus diperiksa berdasarkan data yang ada. Teknik pembelajaran mesin atau *Machine Learning* pun digunakan sebagai alat bantu melakukan diagnosis medis. Pada penelitian ini, penulis membandingkan akurasi dari algoritma *AdaBoost* dengan algoritma *LightGBM* dalam mengklasifikasikan penyakit diabetes dengan tujuan untuk mengetahui algoritma mana yang memiliki tingkat akurasi paling tinggi dan mengetahui evaluasi model dari masing-masing pemodelan. Untuk *dataset* yang digunakan adalah *dataset Pima Indians Diabetes Database* yang memiliki jumlah data sebanyak 768 data. Dari hasil penelitian tersebut, didapatkan bahwa algoritma *LightGBM* memiliki nilai akurasi yang lebih tinggi dibandingkan dengan algoritma *AdaBoost* dengan nilai akurasi *LightGBM* sebesar 91,67% dan nilai AUC sebesar 0,9704 dibandingkan dengan algoritma *AdaBoost* yang memiliki nilai akurasi sebesar 91,14% dan nilai AUC sebesar 0,9693.

**Kata Kunci:** *Data Mining, AdaBoost, LightGBM, Klasifikasi, Pima Indians Diabetes Database*

---

## Abstract

*Diabetes is known as an epidemic disease that affects almost every country, age group, and economic factor around the world. In Indonesia, based on data from the IDF in 2020, of the total adult population of 172 million people, 10.5 million people suffer from diabetes. When designing software or medical diagnosis software, disease prediction becomes one of the most time-consuming tasks because there are many aspects, each of which must be examined based on the available data. Machine learning techniques are also used as a tool to make medical diagnoses. In this study, the authors compare the accuracy of the AdaBoost algorithm with the LightGBM algorithm in classifying diabetes to find out which algorithm has the highest level of accuracy and to know the model evaluation of each model. For the dataset used is the Pima Indians Diabetes Database dataset which has a total of 768 data. From the results of this study, it was found that the LightGBM algorithm has a higher accuracy value than the AdaBoost algorithm with an accuracy value of 91.67% LightGBM and an AUC value of 0.9704 compared to the AdaBoost algorithm which has an accuracy value of 91.14% and an AUC value of 0.9693.*

**Keywords:** *Data Mining, AdaBoost, LightGBM, Classification, Pima Indians Diabetes Database*

---

## 1. Pendahuluan

Secara umum, Diabetes dikenal sebagai penyakit epidemi yang berdampak di hampir setiap negara, kelompok usia, dan faktor ekonomi di seluruh dunia. Berdasarkan data dari *International Diabetes Federation* (IDF) tahun 2015, diperkirakan sebanyak 415 juta jiwa mengidap penyakit diabetes. Angka ini diperkirakan juga

dapat meningkat hingga 640 juta pada tahun 2040 [1]. Selain itu, persentase kematian penderita diabetes sebelum usia 70 tahun lebih tinggi terjadi di negara-negara yang memiliki penghasilan rendah dan menengah dibandingkan dengan negara-negara berpenghasilan tinggi [2].

Di Indonesia sendiri, berdasarkan data dari IDF tahun 2020, dari total populasi orang dewasa sebanyak 172 juta jiwa, 10,5 juta jiwa mengidap

diabetes. prevalensi penderita Diabetes tertinggi berdasarkan status pendidikan merupakan tamatan pendidikan setingkat D1/D2/D3/PT, dan untuk status pekerjaan, yang paling banyak mengidap Diabetes adalah masyarakat yang berstatus sebagai PNS/TNI/Polri/BUMN/BUMD. Rentang penderita Diabetes tersebut paling banyak berada pada usia 55-64 tahun dan 65-74 tahun. Data tersebut didapat dari hasil klasifikasi pemeriksaan gula darah pada Riskesdas 2018 [2]. Kementerian Kesehatan telah melakukan beberapa usaha untuk mengendalikan diabetes seperti membentuk 13.500 Pos Pembinaan Terpadu (Posbindu) agar memudahkan warga untuk melakukan *Medical Diagnosis* atau *Diagnosis Medis*. (Kementerian Kesehatan, 2018). Ketika merancang *software* atau perangkat lunak diagnosis medis, prediksi penyakit menjadi salah satu tugas yang memakan banyak waktu dikarenakan terdapat banyak aspek yang masing-masing harus diperiksa berdasarkan data yang ada. Teknik pembelajaran mesin atau *Machine Learning* pun digunakan sebagai alat bantu melakukan diagnosis medis [3].

*Machine Learning* adalah salah satu domain penting dalam penelitian yang memiliki tujuan untuk memprediksi dan melakukan tinjauan sistematis [3]. Terdapat tiga kelas pembelajaran pada *Machine Learning* yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning* [4].

*Supervised Learning* digunakan untuk melakukan klasifikasi dan prediksi keluaran nilai data, *Unsupervised Learning* digunakan untuk mendapatkan pola pada sekumpulan data atau yang biasa disebut *clustering*, dan yang terakhir *Reinforcement Learning* digunakan untuk mempelajari suatu kebijakan atau membuat strategi di suatu lingkungan yang ditentukan untuk mendapatkan hasil yang paling baik [4].

*Machine learning* berbasis *supervised learning* dapat digunakan untuk mengklasifikasikan dan mendeteksi penyakit diabetes. Beberapa penelitian sebelumnya telah dilakukan, salah satunya adalah penelitian yang dilakukan oleh Deepti Sisodia dan Dilip Singh Sisodia dengan judul *Prediction of Diabetes using Classification Algorithms*. Penelitian tersebut memprediksi diabetes dengan menggunakan algoritma

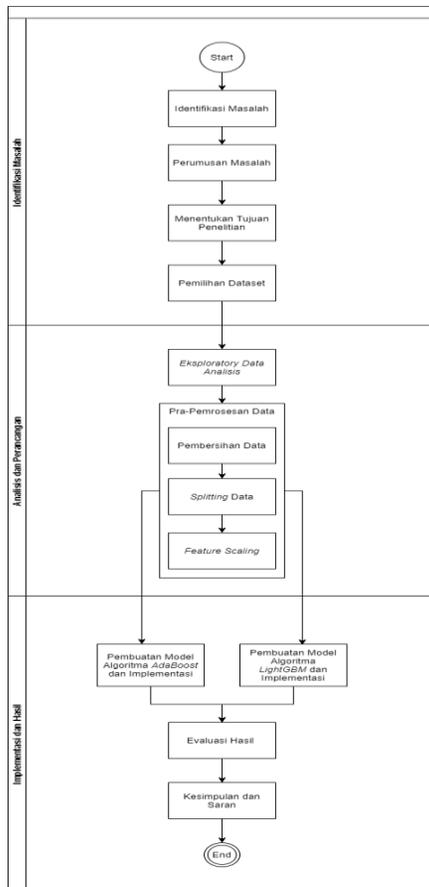
klasifikasi *Naive Bayes*, *Support Vector Machine* (SVM) dan *Decision Tree* dengan hasil akurasi masing-masing algoritma 76.3% untuk *Naive Bayes*, 65.1% untuk SVM, dan 73.82% untuk *Decision Tree* [5].

Dari penelitian tersebut, algoritma *AdaBoost* dan algoritma *LightGBM* belum digunakan sebagai metode pengklasifikasian diabetes. Selain itu, kedua algoritma tersebut dikenal memiliki tingkat akurasi yang tinggi, seperti pada penelitian yang dilakukan oleh Agustin Yoga Handoko, Kusri, dan Luthfi Emha Taufiq dengan judul “Klasifikasi Penerimaan Mahasiswa Baru Menggunakan Algoritma C4.5 Dan *Adaboost*” tingkat akurasi algoritma *AdaBoost* memiliki hasil yang sama dengan algoritma C4.5 sebesar 90.28% (Agustin et al., 2017) dan penelitian yang dilakukan oleh Wang Bijun, Wang Yulong, Qin Kun, Xia Qizhi dengan judul “*Detecting Transportation Modes Based on LightGBM Classifier from GPS Trajectory Data*” yang membandingkan akurasi dari algoritma *Decision Tree*, *LightGBM*, dan *XGBoost*. Algoritma *LightGBM* unggul dengan tingkat akurasi sebesar 90.61% [6].

Berdasarkan permasalahan dan penelitian yang telah ada sebelumnya, penulis akan melakukan penelitian yang membandingkan tingkat akurasi Algoritma *AdaBoost* dengan Algoritma *LightGBM* secara spesifik dalam mengklasifikasikan penyakit diabetes.

## 2. Metode Penelitian

Untuk sistematika penyelesaian masalah pada penelitian ini, penulis membagi menjadi tiga tahapan umum, tahapan-tahapan tersebut adalah tahap identifikasi masalah, analisis dan perancangan, dan implementasi dan hasil.



Gambar 1 Sistematika Penyelesaian Masalah

- Pada tahap identifikasi masalah, penulis menentukan rumusan masalah yang ada berdasarkan masalah yang sudah teridentifikasi. Selanjutnya penulis menentukan tujuan dari penelitian yang akan dilaksanakan berdasarkan rumusan masalah yang telah dirumuskan. Langkah terakhir pada tahapan identifikasi masalah adalah pemilihan *dataset* yang sesuai dengan rumusan masalah dan tujuan dari penelitian.
- Pada tahap analisis dan perancangan, penulis melakukan EDA untuk mencari informasi yang ada pada *dataset* yang digunakan [7]. Selanjutnya penulis melakukan pra-pemrosesan data yang tahapannya adalah pembersihan data, *splitting* data dan diakhiri dengan *feature scaling* [8].
- Pada tahap implementasi dan hasil, penulis membuat pemodelan dari algoritma *AdaBoost* dan *LightGBM* yang dilanjutkan dengan implementasi dari masing-masing algoritma. *Adaboost* adalah algoritma yang

melakukan klasifikasi berdasarkan algoritma dasar seperti *decision tree*, *K-nearest neighbors* (KNN), dll. *Classifier* pada algoritma ini diatur agar mendukung sampel yang salah diklasifikasikan. Algoritma ini menggunakan *database* berulang kali dan *database* yang digunakan tidak harus besar [9]. Selain itu algoritma ini meningkatkan performa memprediksi dari beberapa *weak classifier* menjadi satu *strong classifier* dengan memfokuskan *data point* [10]. *LightGBM* adalah *framework* pembelajaran gradien berbasis pembelajaran pohon. Algoritma ini menggunakan basis histogram untuk mempercepat proses *train*, mengurangi konsumsi memori, dan mengadopsi strategi *leaf-wise leaf growth strategy* dengan limitasi yang dalam [11]. Setelah membuat dan mengimplementasikan masing-masing algoritma, penulis mengevaluasi hasil dan membuat kesimpulan dan saran.

Pada penelitian ini, data yang digunakan merupakan data *Pima Indians Diabetes Database* yang diambil dari Kaggle. Atribut informasi yang ada pada PIDD terdiri-dari: *Pregnancies*, *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, *BMI*, *DiabetesPedigree*, *Age*, *Outcome*. PIDD memiliki 2 *class* yaitu: 0 untuk pasien yang dinyatakan negatif diabetes atau sehat dan 1 untuk pasien yang dinyatakan positif diabetes.

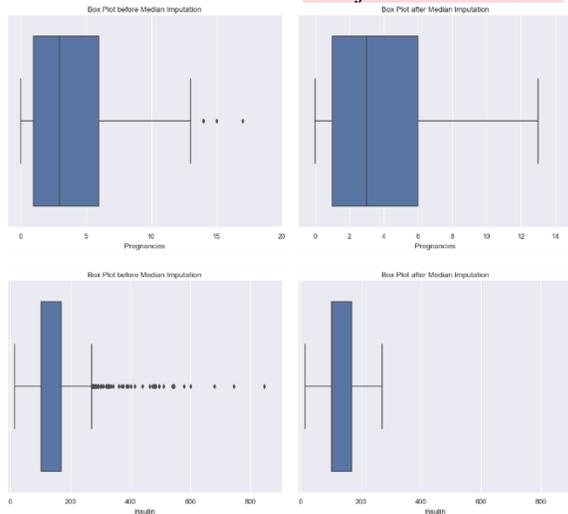
Pada tahap pengolahan data, peneliti menggunakan metode *AdaBoost* dan *LightGBM* yang digunakan untuk membuat model prediksinya. Sedangkan untuk pengujian model prediksi klasifikasi data pada penelitian ini, penulis menggunakan *Python* sebagai bahasa pemrograman dan menggunakan *Jupyter Notebook* sebagai *software* yang digunakan. *Python* adalah salah satu bahasa pemrograman yang mudah dipelajari dan sering digunakan dalam pemrograman. Bahasa pemrograman ini memiliki struktur data tingkat tinggi yang efisien dan pendekatan yang sederhana namun efektif untuk pemrograman berorientasi objek [12]. Hasil klasifikasi dari data pengujian akan dibandingkan untuk melihat hasil akurasi terbaik.

Metode evaluasi pada penelitian ini adalah membandingkan tingkat akurasi antara algoritma *Adaboost* dengan *LightGBM*. Penulis juga dapat

mengetahui *confusion matrix* dan nilai dari *precision, recall, F1-Score* dan ROC AUC untuk mengetahui performa dari masing-masing algoritma tersebut.

**3. Hasil dan Pembahasan**

Penelitian ini diawali dengan tahapan *pre processing* data. Pada tahapan tersebut, penulis mengubah nilai 0 pada atribut menjadi nilai median dari masing-masing atribut. Beberapa atribut yang diubah adalah *Glucose, BloodPressure, SkinThickness, Insulin,* dan *BMI*. Setelah mengubah nilai nol pada atribut tersebut, penulis menghilangkan *outliers* agar hasil dari klasifikasi data diabetes menjadi lebih akurat



Gambar 2 Box Plot Median Imputation

Dapat dilihat pada Gambar 2, atribut *Pregnancies* dan *Insulin* di sebelah kiri sebelum *outliers* dihilangkan, data masih tersebar dan memiliki *range* yang luas. Setelah *outliers* dihilangkan (Gambar 2 sebelah kanan) range pada atribut *pregnancies* dan *insulin* pun menurun. Langkah ini dilakukan pada semua atribut yang ada sehingga semua *outliers* pada masing-masing atribut hilang dan data siap digunakan. Untuk rasio data *train* dan data testing yang digunakan, penulis membandingkan beberapa rasio untuk menemukan rasio mana yang paling tepat pada penelitian ini. Beberapa rasio yang digunakan dapat dilihat pada Tabel 1.

Tabel 1 Splitting Data Test Dan Train

Train	Testing	Jumlah Data Train	Jumlah Data Testing
70%	30%	537	231
75%	25%	576	192

80%	20%	614	154
-----	-----	-----	-----

Pada penelitian ini, penulis akan membandingkan algoritma *AdaBoost* dengan *LightGBM* dengan menggunakan tiga perbandingan rasio. Untuk perbandingan rasio yang pertama adalah 70:30 dengan jumlah data *train* sebanyak 537 dan jumlah data testing sebanyak 231. Perbandingan rasio yang kedua adalah 75:25 dengan jumlah data *train* sebanyak 576 dan data testing sebanyak 192. Dan yang terakhir adalah perbandingan rasio 80:20 dengan jumlah data *train* sebanyak 614 dan data testing sebanyak 154. Pada tahapan ini, peneliti melakukan *feature scaling* agar setiap atribut pada data *training* yang ada memiliki rentang yang tidak jauh berbeda dengan atribut yang lain.

```
array([[ 9. , 89. , 62. , ..., 22.5 , 0.142, 33. ],
       [ 1. , 118. , 58. , ..., 33.3 , 0.261, 23. ],
       [ 0. , 91. , 80. , ..., 32.4 , 0.601, 27. ],
       ...,
       [ 4. , 94. , 65. , ..., 24.7 , 0.148, 21. ],
       [ 11. , 85. , 74. , ..., 30.1 , 0.3 , 35. ],
       [ 5. , 136. , 82. , ..., 30.1 , 0.64 , 29. ]])
```

Gambar 3 Data Train Sebelum Feature Scaling

Dapat dilihat pada Gambar 3 terdapat perbedaan rentang nilai yang jauh antara nilai pada atribut *Pregnancies* (nilai *array* pertama dari sebelah kiri) dengan nilai *DiabetesPedigreeFunction* (nilai *array* kedua dari sebelah kanan).

```
array([[ 1.59270295e+00, -1.09892515e+00, -9.60403972e-01, ...,
       -1.51558155e+00, -1.16441385e+00,  5.17103288e-03],
       [-8.37487513e-01, -1.33609327e-01, -1.32563806e+00, ...,
       1.36867329e-01, -6.72608694e-01, -8.97409251e-01],
       [-1.14126132e+00, -1.03235165e+00,  6.83149446e-01, ...,
       -8.36743499e-04,  7.32548882e-01, -5.36377137e-01],
       ...,
       [ 7.38339118e-02, -9.32491390e-01, -6.86478402e-01, ...,
       -1.17897159e+00, -1.13961695e+00, -1.07792531e+00],
       [ 2.20025057e+00, -1.23207216e+00,  1.35298307e-01, ...,
       -3.52747152e-01, -5.11428854e-01,  1.85687090e-01],
       [ 3.77607720e-01,  4.65552221e-01,  8.65766492e-01, ...,
       -3.52747152e-01,  8.93728721e-01, -3.55861081e-01]])
```

Gambar 4 Data Train Setelah Feature Scaling

Dari Gambar 4 dapat dilihat bahwa setelah dilakukan *feature scaling*, rentang nilai antara atribut *Pregnancies* (nilai *array* pertama dari sebelah kiri) dengan atribut *DiabetesPedigreeFunction* (nilai *array* kedua dari sebelah kanan) tidak jauh berbeda. Data yang telah dilakukan pra-pemrosesan selanjutnya diimplementasikan ke dalam pemodelan algoritma. Untuk pemodelan algoritma yang digunakan adalah algoritma *AdaBoost* dan *LightGBM*.

Tabel 2 Hasil Percobaan Awal

Train	Testing	Akurasi Adaboost	Akurasi LightGBM
70%	30%	86.58%	88.31%
75%	25%	90.10%	90.10%

80%	20%	85.71%	89.61%
-----	-----	--------	--------

Dapat dilihat pada Tabel 2, dari percobaan awal menggunakan beberapa perbandingan rasio, didapatkan tingkat akurasi dalam mengklasifikasikan penderita penyakit diabetes yang paling tinggi adalah klasifikasi dengan menggunakan perbandingan rasio 75% untuk data *training* dan 25% untuk data testing. Oleh karena itu, dalam penelitian ini penulis menggunakan perbandingan rasio 75% untuk data *training* dan 25% untuk data testing.

Tabel 3 Hasil Implementasi Awal

Model	Accuracy	K-Fold Accuracy	Precision	Recall	F1
LightGBM	90.1042	87.3170	0.8209	0.8871	0.8527
AdaBoost	90.1042	85.7502	0.8209	0.8871	0.8527

Dengan menggunakan perbandingan 75:25, didapatkan hasil akurasi sebesar 90.10417 untuk algoritma *LightGBM* dan 90.10417 untuk algoritma *AdaBoost* dengan rata-rata akurasi *K-fold* 87.316999 untuk *LightGBM* dan 85.750151 untuk *AdaBoost*. Masing-masing algoritma tersebut memiliki nilai *precision* dan *recall* yang sama sehingga keduanya memiliki *F1 Score* yang sama juga. Untuk nilai dari *precision*, *recall*, dan *F1 Score* dapat dilihat pada Tabel 3.

Untuk mendapatkan tingkat akurasi yang lebih tinggi, penulis melakukan *tuning* pada masing-masing algoritma. Setiap algoritma diatur menggunakan parameter terbaik yang telah dibandingkan sebelumnya.

```

Fitting 10 folds for each of 20 candidates, totalling 200 fits
AdaBoostClassifier():
Best Accuracy : 86.79%
Best Parameters : {'learning_rate': 0.1, 'n_estimators': 200, 'random_state': 0}

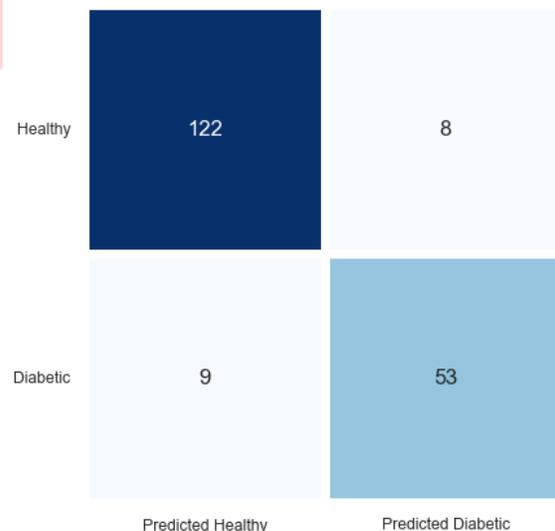
Fitting 10 folds for each of 20 candidates, totalling 200 fits
LGBMClassifier():
Best Accuracy : 87.49%
Best Parameters : {'learning_rate': 0.1, 'n_estimators': 200, 'random_state': 0}
    
```

Gambar 5 Hasil Tuning Hyperparameter

Tuning parameter yang telah dilakukan adalah sebanyak 10 kali *folds*. Masing-masing algoritma memiliki kandidat berjumlah 20 dengan total 200 *fits*. Dari 200 *fits* tersebut menghasilkan parameter terbaik untuk algoritma *AdaBoost* dan *LightGBM* yang hasilnya dapat dilihat pada Gambar V.1. Dari hasil tersebut, penulis mengimplementasikan parameter tersebut ke masing-masing algoritma dengan metode *boosting* yang berbeda.

Setelah melakukan *tuning hyperparameter*, hasil parameter yang ada pada Gambar 5 penulis implementasikan ke masing-masing algoritma. Untuk algoritma *AdaBoost*, penulis menggunakan dua metode *boosting* SAMME, dan SAMME.R. Sedangkan untuk *LightGBM*, penulis menggunakan metode *boosting Traditional Gradient Boosting Decision Tree (GBDT)*, *Dropouts Meet Multiple Additive Regression Trees (DART)*, dan *Gradient-Based One-Side Sampling (GOSS)*.

Hasil implementasi algoritma *AdaBoost* menggunakan metode *boosting* SAMME menghasilkan jumlah data *True Negative (TN)* sebanyak 122, *False Negative (FN)* sebanyak 9, *True Positive (TP)* sebanyak 53, *False Positive (FP)* sebanyak 8 data. Untuk detail dari *Confusion Matrix* *AdaBoost* menggunakan metode *boosting* SAMME dapat dilihat pada Gambar 6.



Gambar 6 Confusion Matrix AdaBoost Menggunakan SAMME

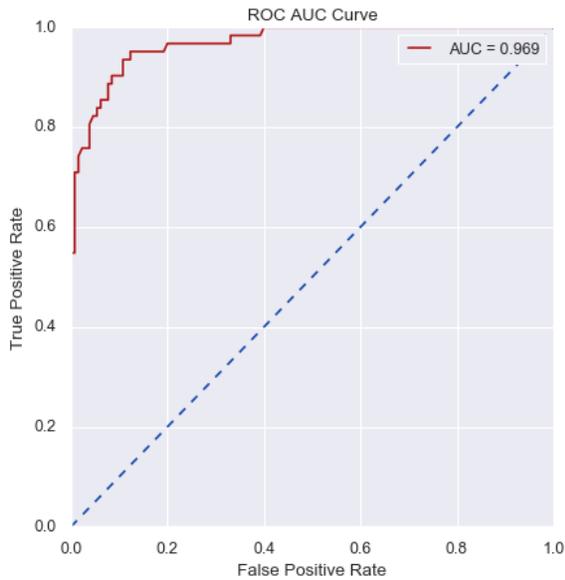
Setelah menemukan nilai dari masing-masing TP, TN, FP, dan FN, penulis dapat menentukan nilai dari *precision*, *recall*, *f1-Score*, dan akurasi. Selain itu, penulis juga dapat mengetahui nilai dari ROC AUC dan rata-rata akurasi untuk *cross validation* dari data yang ada. Untuk nilai dari masing-masing *precision*, *recall*, *f1-score*, akurasi, ROC AUC, dan rata-rata akurasi *cross validation* dapat dilihat pada Tabel 4.

Tabel 4 Hasil Pengujian Pemodelan AdaBoost Menggunakan Algoritma Boosting SAMME

	Precision	Recall	F1-Score	Support
0 (Healthy)	0.93	0.94	0.93	130
1 (Diabetes)	0.87	0.85	0.86	62
Accuracy			0.91	192

<b>Macro Avg</b>	0.9	0.9	0.9	192
<b>Weighted Avg</b>	0.91	0.91	0.91	192
<hr/>				
<b>ROC AUC score</b>	0.9693			
<b>Accuracy score</b>	0.9114			
<b>CV Mean Accuracy</b>	84.35%			

Dapat diketahui juga bentuk kurva dari *score* ROC AUC pada Algoritma *AdaBoost* yang menggunakan metode *boosting* SAMME dalam Gambar 7.



Gambar 7 Kurva ROC AUC *AdaBoost* Menggunakan *Boosting* SAMME

Hasil implementasi Algoritma *AdaBoost* menggunakan metode *boosting* SAMME.R menghasilkan jumlah data *True Negative (TN)* sebanyak 119, *False Negative (FN)* sebanyak 9, *True Positive (TP)* sebanyak 53, *False Positive (FP)* sebanyak 11 data. Untuk detail dari *Confusion Matrix* *AdaBoost* menggunakan metode *boosting* SAMME.R dapat dilihat pada Gambar 8.

Healthy	119	11
Diabetic	9	53
	Predicted Healthy	Predicted Diabetic

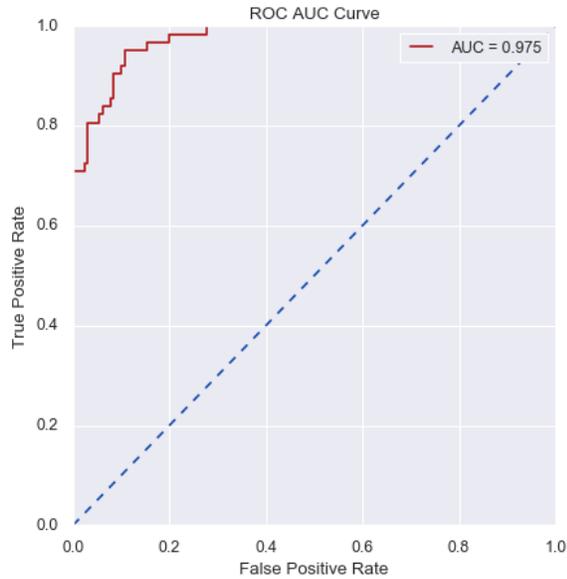
Gambar 8 *Confusion Matrix* *AdaBoost* Menggunakan SAMME.R

Setelah menemukan nilai dari masing-masing TP, TN, FP, dan FN, penulis dapat menentukan nilai dari *precision*, *recall*, *f1-Score*, dan akurasi. Selain itu, penulis juga dapat mengetahui nilai dari ROC AUC dan rata-rata akurasi untuk *cross validation* dari data yang ada. Untuk nilai dari masing-masing *precision*, *recall*, *f1-score*, akurasi, ROC AUC, dan rata-rata akurasi *cross validation* dapat dilihat pada Tabel 5.

Tabel 5 Hasil Pengujian Pemodelan *AdaBoost* Menggunakan Algoritma *Boosting* SAMME.R

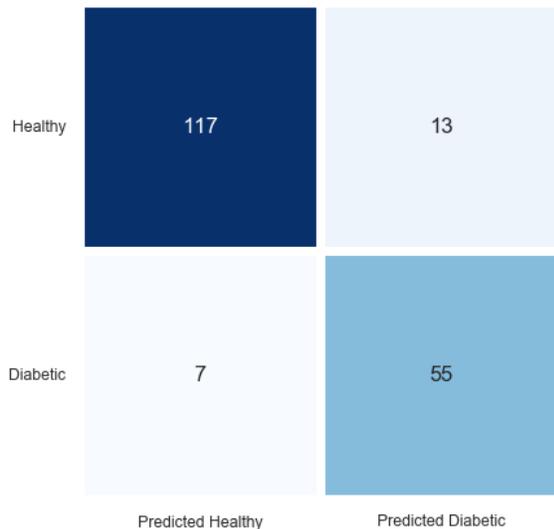
	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>0 (Healthy)</b>	0.93	0.92	0.92	130
<b>1 (Diabetes)</b>	0.83	0.85	0.84	62
<hr/>				
<b>Accuracy</b>			0.9	192
<b>Macro Avg</b>	0.88	0.89	0.88	192
<b>Weighted Avg</b>	0.9	0.9	0.9	192
<hr/>				
<b>ROC AUC score</b>	0.9746			
<b>Accuracy score</b>	0.8958			
<b>CV Mean Accuracy</b>	86.79%			

Dapat diketahui juga bentuk kurva dari *score* ROC AUC pada Algoritma *AdaBoost* yang menggunakan metode *boosting* SAMME.R dalam Gambar 9.



Gambar 9 Kurva ROC AUC AdaBoost Menggunakan Boosting SAMME.R

Hasil implementasi algoritma *LightGBM* menggunakan *Traditional Gradient Boosting Decision Tree* menghasilkan jumlah data *True Negative (TN)* sebanyak 117, *False Negative (FN)* sebanyak 7, *True Positive (TP)* sebanyak 55, *False Positive (FP)* sebanyak 13 data. Untuk detail dari *Confusion Matrix LightGBM* menggunakan *Traditional Gradient Boosting Decision Tree* dapat dilihat pada Gambar 10.



Gambar 10 Confusion Matrix LightGBM Menggunakan Traditional Gradient Boosting Decision Tree

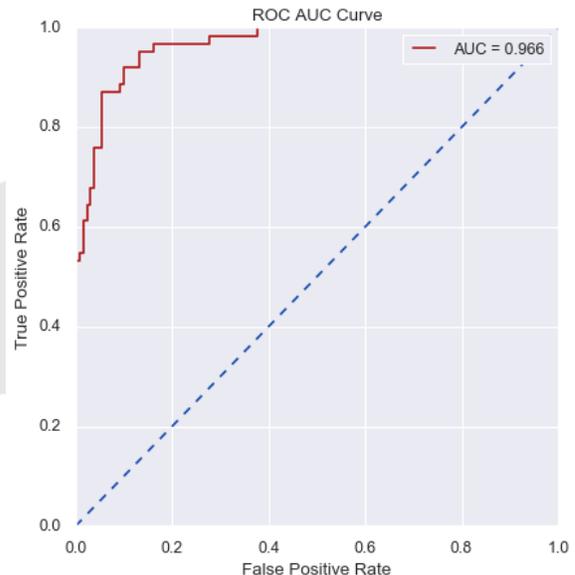
Setelah menemukan nilai dari masing-masing TP, TN, FP, dan FN, penulis dapat menentukan nilai dari

*precision*, *recall*, *f1-Score*, dan akurasi. Selain itu, penulis juga dapat mengetahui nilai dari ROC AUC dan rata-rata akurasi untuk *cross validation* dari data yang ada. Untuk nilai dari masing-masing *precision*, *recall*, *f1-score*, akurasi, ROC AUC, dan rata-rata akurasi *cross validation* dapat dilihat pada Tabel 6.

Tabel 6 Hasil Pengujian Pemodelan *LightGBM* Menggunakan *Traditional Gradient Boosting Decision Tree*

	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>0 (healthy)</b>	0.94	0.9	0.92	130
<b>1 (Diabetes)</b>	0.81	0.89	0.85	62
<b>Accuracy</b>			0.9	192
<b>Macro Avg</b>	0.88	0.89	0.88	192
<b>Weighted Avg</b>	0.9	0.9	0.9	192
<b>ROC AUC score</b>	0.9658			
<b>Accuracy score</b>	0.8958			
<b>CV Mean Accuracy</b>	87.49%			

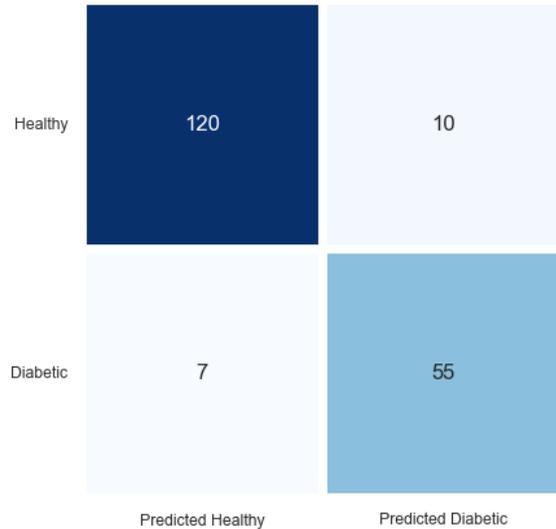
Dapat diketahui juga bentuk kurva dari *score ROC AUC* pada Algoritma *LightGBM* yang menggunakan metode *boosting GBDT* dalam Gambar 11.



Gambar 11 Kurva ROC AUC LightGBM Menggunakan Boosting GBDT

Hasil implementasi algoritma *LightGBM* menggunakan *Dropout Meet Multiple Additive*

Regression Trees menghasilkan jumlah data True Negative (TN) sebanyak 120, False Negative (FN) sebanyak 7, True Positive (TP) sebanyak 55, False Positive (FP) sebanyak 10 data. Untuk detail dari Confusion Matrix LightGBM menggunakan Dropout Meet Multiple Additive Regression Trees dapat dilihat pada Gambar 12.



Gambar 12 Confusion Matrix LightGBM Menggunakan Dropout Meet Multiple Additive Regression Trees

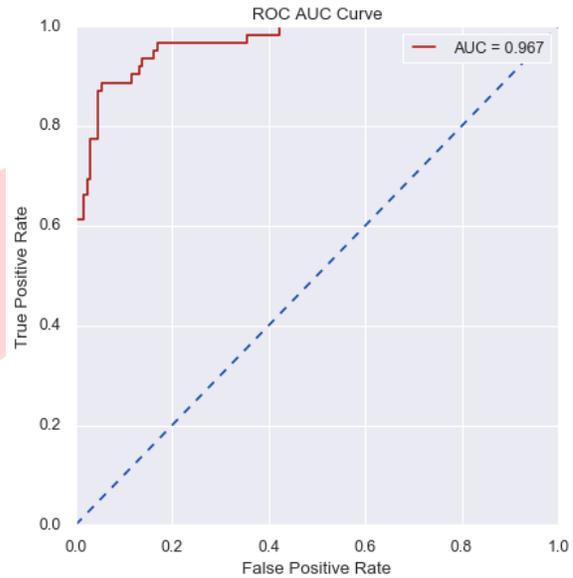
Setelah menemukan nilai dari masing-masing TP, TN, FP, dan FN, penulis dapat menentukan nilai dari precision, recall, f1-Score, dan akurasi. Selain itu, penulis juga dapat mengetahui nilai dari ROC AUC dan rata-rata akurasi untuk cross validation dari data yang ada. Untuk nilai dari masing-masing precision, recall, f1-score, akurasi, ROC AUC, dan rata-rata akurasi cross validation dapat dilihat pada Tabel 7.

Tabel 7 Hasil Pengujian Pemodelan LightGBM Menggunakan Dropout Meet Multiple Additive Regression Trees

	Precision	Recall	F1-Score	Support
0 (healthy)	0.94	0.92	0.93	130
1 (diabetes)	0.85	0.89	0.87	62
Accuracy			0.91	192
Macro Avg	0.9	0.91	0.9	192
Weighted Avg	0.91	0.91	0.91	192
ROC AUC score	0.9667			
Accuracy score	0.9114			

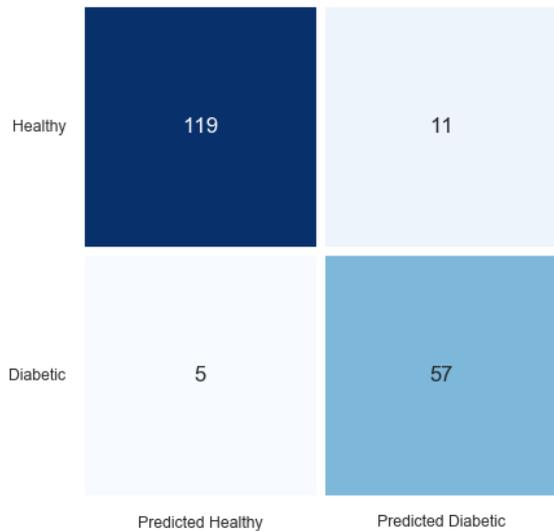
	Precision	Recall	F1-Score	Support
CV Mean Accuracy	86.45 %			

Dapat diketahui juga bentuk kurva dari score ROC AUC pada algoritma LightGBM yang menggunakan metode boosting DART dalam Gambar 13.



Gambar 13 Kurva ROC AUC LightGBM Menggunakan Boosting DART

Hasil implementasi algoritma LightGBM menggunakan Gradient-Based One-Side Sampling menghasilkan jumlah data True Negative (TN) sebanyak 119, False Negative (FN) sebanyak 5, True Positive (TP) sebanyak 57, False Positive (FP) sebanyak 11 data. Untuk detail dari Confusion Matrix LightGBM menggunakan Gradient-Based One-Side Sampling dapat dilihat pada Gambar 14.



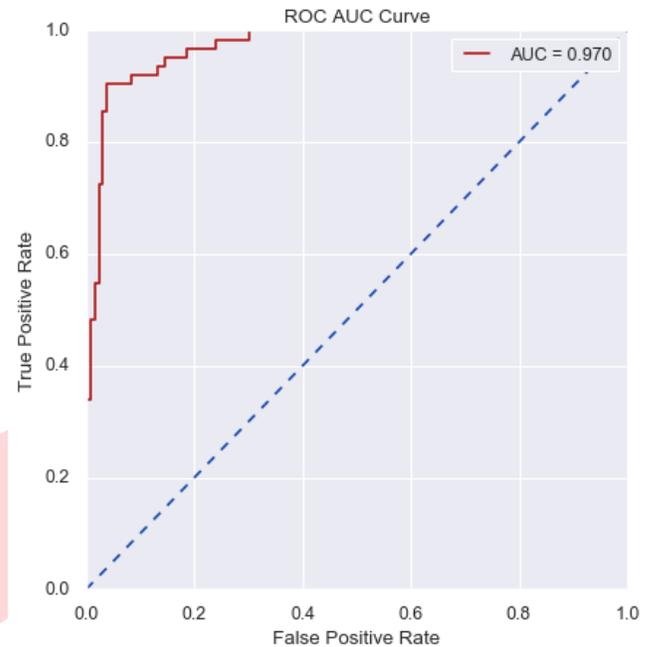
Gambar 14 Confusion Matrix LightGBM Menggunakan Gradient-Based One-Side Sampling

Setelah menemukan nilai dari masing-masing TP, TN, FP, dan FN, penulis dapat menentukan nilai dari *precision*, *recall*, *f1-Score*, dan akurasi. Selain itu, penulis juga dapat mengetahui nilai dari ROC AUC dan rata-rata akurasi untuk *cross validation* dari data yang ada. Untuk nilai dari masing-masing *precision*, *recall*, *f1-score*, akurasi, ROC AUC, dan rata-rata akurasi *cross validation* dapat dilihat pada Tabel 8.

Tabel 8 Hasil Pengujian Pemodelan LightGBM Menggunakan Gradient-Based One-Side Sampling

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<b>0 (healthy)</b>	0.96	0.92	0.94	130
<b>1 (Diabetes)</b>	0.84	0.92	0.88	62
<b>Accuracy</b>			0.92	192
<b>Macro Avg</b>	0.9	0.92	0.91	192
<b>Weighted Avg</b>	0.92	0.92	0.92	192
<b>ROC AUC score</b>	0.9704			
<b>Accuracy score</b>	0.9167			
<b>CV Mean Accuracy</b>	85.76%			

Dapat diketahui juga bentuk kurva dari *score* ROC AUC pada Algoritma LightGBM yang menggunakan metode *boosting* GOSS dalam Gambar 15.



Gambar 15 Kurva ROC AUC LightGBM Menggunakan Boosting GOSS

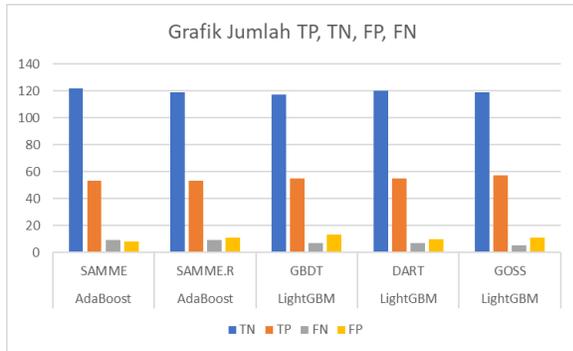
Dari hasil implementasi tiap algoritma dengan metode *boosting* yang berbeda, penulis selanjutnya membandingkan jumlah nilai TP, TN, FP, dan FN dari masing-masing algoritma.

Tabel 9 Jumlah Nilai TP, TN, FP, FN Dari Masing-Masing Algoritma

	<i>AdaBoost</i>	<i>AdaBoost</i>	<i>LightGBM</i>	<i>LightGBM</i>	<i>LightGBM</i>
	<i>SAMME</i>	<i>SAMME.R</i>	<i>GBDT</i>	<i>DART</i>	<i>GOSS</i>
<b>TN</b>	122	119	117	120	119
<b>TP</b>	53	53	55	55	57
<b>FN</b>	9	9	7	7	5
<b>FP</b>	8	11	13	10	11

Dari Tabel 9, dapat diketahui bahwa algoritma dengan jumlah TN paling banyak terdapat pada algoritma *AdaBoost* yang menggunakan metode *boosting* SAMME dengan jumlah 122 TN diikuti dengan algoritma *LightGBM* yang menggunakan metode *boosting* DART dengan jumlah 120 TN. Untuk algoritma dengan jumlah TP paling banyak terdapat pada algoritma *LightGBM* yang menggunakan metode *boosting* GOSS dengan jumlah 57 TP diikuti oleh algoritma *LightGBM* yang menggunakan metode DART dan GBDT dengan jumlah yang sama sebanyak 55 TP. Untuk algoritma dengan jumlah FN paling

sedikit adalah algoritma *LightGBM* yang menggunakan metode *boosting* GOSS dengan jumlah 5 FN diikuti algoritma *LightGBM* yang menggunakan metode *boosting* DART dan GBDT dengan jumlah FN yang sama sebanyak 7 FN. Untuk jumlah FP paling sedikit adalah algoritma *AdaBoost* yang menggunakan metode *boosting* SAMME sebanyak 8 FP diikuti dengan algoritma *LightGBM* yang menggunakan metode *boosting* DART sebanyak 10 FP. Penulis juga memvisualisasikan Tabel 9 ke dalam bentuk grafik agar memudahkan pembaca.



Gambar 16 Grafik jumlah TP, TN, FP, FN

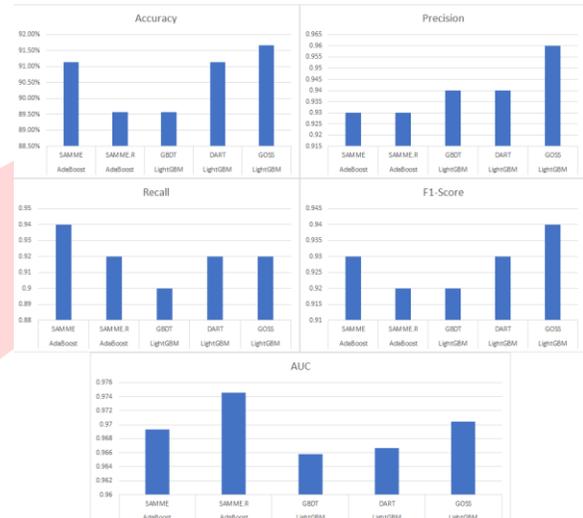
Selanjutnya penulis juga membandingkan performa dari masing-masing algoritma. Detail performa dari masing-masing algoritma dapat dilihat pada Tabel 10.

Tabel 10 Komparasi Perbandingan Hasil Evaluasi

	<i>AdaBoos</i>	<i>AdaB</i>	<i>Light</i>	<i>Light</i>	<i>Light</i>
	<i>t</i>	<i>oost</i>	<i>GBM</i>	<i>GBM</i>	<i>GBM</i>
	<i>SAM</i>	<i>SAM</i>	<i>GBD</i>	<i>DAR</i>	<i>GOSS</i>
	<i>ME</i>	<i>ME.R</i>	<i>T</i>	<i>T</i>	
<b>Accu</b>	91.14	89.58	89.58	91.14	91.67
<b>racy</b>	%	%	%	%	%
<b>Preci</b>	0.93	0.93	0.94	0.94	0.96
<b>sion</b>					
<b>Reca</b>	0.94	0.92	0.9	0.92	0.92
<b>ll</b>					
<b>FI-</b>	0.93	0.92	0.92	0.93	0.94
<b>Scor</b>					
<b>e</b>					
<b>AUC</b>	0.969	0.974	0.965	0.966	0.970
	3	6	8	7	4

Dari Tabel 10, dapat diketahui bahwa algoritma *LightGBM* yang menggunakan metode *boosting* GOSS memiliki persentase akurasi paling tinggi dibanding dengan algoritma lain dengan tingkat akurasi sebesar 91.67% diikuti oleh algoritma *LightGBM* yang menggunakan metode *boosting* DART dan *AdaBoost* yang menggunakan metode *boosting* SAMME dengan tingkat akurasi yang sama

sebesar 91.14%. Untuk performa algoritma paling baik adalah algoritma *AdaBoost* yang menggunakan metode *boosting* SAMME.R dengan nilai AUC sebesar 0.9746 diikuti dengan algoritma *LightGBM* yang menggunakan metode *boosting* GOSS dengan nilai AUC sebesar 0.9704. Perbandingan nilai *Accuracy*, *Precision*, *Recall*, *F-1 Score*, dan *AUC* dalam bentuk grafik dapat dilihat pada Gambar 12.



Gambar 17 Hasil Evaluasi Masing-Masing Algoritma

#### 4. Kesimpulan

Dari penelitian tentang perbandingan akurasi algoritma *AdaBoost* dan *LightGBM* dalam mengklasifikasikan penyakit diabetes, dapat disimpulkan bahwa:

Metode *boosting* yang memiliki tingkat akurasi paling tinggi pada algoritma *AdaBoost* adalah metode *boosting* SAMME dengan nilai akurasi sebesar 91.14% dibandingkan dengan algoritma *AdaBoost* yang menggunakan metode *boosting* SAMME.R yaitu hanya sebesar 89.58%. Algoritma *AdaBoost* dengan metode SAMME memiliki nilai *precision* sebesar 0.93 dan *recall* sebesar 0.94. Dari nilai *precision* dan *recall* tersebut, didapatkan hasil f1-Score sebesar 0.93 dan nilai AUC sebesar 0.9693.

Metode *boosting* yang memiliki tingkat akurasi paling tinggi pada algoritma *LightGBM* adalah metode *boosting* GOSS dengan nilai akurasi sebesar 91.67% dibandingkan dengan algoritma *LightGBM* yang menggunakan metode *boosting* DART dengan akurasi sebesar 91.14% dan algoritma *LightGBM* yang menggunakan metode *boosting* GBDT dengan akurasi sebesar 89.58%. Algoritma *LightGBM* dengan metode *boosting* GOSS memiliki nilai *precision* sebesar 0.96,

dan *recall* sebesar 0.92. Dari nilai *precision* dan *recall* tersebut, didapatkan hasil f1-score sebesar 0.94 dan AUC sebesar 0.9704.

Dari hasil penelitian ini, didapatkan bahwa tingkat akurasi dari algoritma *LightGBM* lebih tinggi daripada algoritma *AdaBoost* dengan tingkat akurasi *LightGBM* yang menggunakan metode *boosting* terbaik sebesar 91.67% dengan nilai AUC sebesar 0.9704 dan algoritma *AdaBoost* yang menggunakan metode *boosting* terbaik sebesar 91.14% dengan AUC sebesar 0.9693.

### Referensi

- [1] K. Papatheodorou, M. Banach, E. Bekiari, M. Rizzo, and M. Edmonds, "Complications of Diabetes 2017," *J. Diabetes Res.*, vol. 2018, 2018, doi: 10.1155/2018/3086167.
- [2] Kementerian Kesehatan, "HARI DIABETES SEDUNIA TAHUN 2018 Definisi Diabetes," pp. 1–10, 2018.
- [3] A. Choudhury and D. Gupta, *A Survey on Medical Diagnosis of Diabetes Using Machine Learning Techniques*, vol. 740. Springer Singapore, 2019.
- [4] G. Carleo *et al.*, "Machine learning and the physical sciences," *Rev. Mod. Phys.*, vol. 91, no. 4, p. 45002, 2019, doi: 10.1103/RevModPhys.91.045002.
- [5] D. Sisodia and D. S. Sisodia, "Prediction of Diabetes using Classification Algorithms," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1578–1585, 2018, doi: 10.1016/j.procs.2018.05.122.
- [6] B. Wang, Y. Wang, K. Qin, and Q. Xia, "Detecting Transportation Modes Based on LightGBM Classifier from GPS Trajectory Data," *Int. Conf. Geoinformatics*, vol. 2018-June, no. 41471326, pp. 1–7, 2018, doi: 10.1109/GEOINFORMATICS.2018.8557149.
- [7] V. Cox, "Translating Statistics to Make Decisions," *Transl. Stat. to Make Decis.*, 2017, doi: 10.1007/978-1-4842-2256-0.
- [8] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD Process for Extracting Useful Knowledge from Volumes of Data," *Commun. ACM*, vol. 39, no. 11, pp. 27–34, 1996, doi: 10.1145/240455.240464.
- [9] S. Bahramian and A. Nikravanshalmani, "Hybrid algorithm based on K-nearest-neighbor algorithm and Adaboost with selection of feature by genetic algorithms for the diagnosis of diabetes," *Int. J. Mechatronics, Electr. Comput. Technol.*, vol. 6, no. 21, pp. 2977–2986, 2016.
- [10] V. Rawat and Suryakant, "A classification system for diabetic patients with machine learning techniques," *Int. J. Math. Eng. Manag. Sci.*, vol. 4, no. 3, pp. 729–744, 2019, doi: 10.33889/IJMEMS.2019.4.3-057.
- [11] S. Yang and H. Zhang, "Comparison of Several Data Mining Methods in Credit Card Default Prediction," *Intell. Inf. Manag.*, vol. 10, no. 05, pp. 115–122, 2018, doi: 10.4236/iim.2018.105010.
- [12] F. Menczer, S. Fortunato, and C. A. Davis, "Python Tutorial," *A First Course Netw. Sci.*, pp. 221–237, 2020, doi: 10.1017/9781108653947.010.