

PERBANDINGAN PERFORMA BAGGING DAN ADABOOST DENGAN BASIS LOGISTIC REGRESSION UNTUK MENGKLASIFIKASI DATA MULTI-CLASS

Osvaldo Figo, Dr. Ir. Samuel Lukas, M.Tech.,
Dion Krisnadi, S.Inf., S.Si., M.T.I., M.Act.Sc.

Ringkasan

Salah satu teknik yang paling populer digunakan untuk meningkatkan performa sebuah algoritma *Machine Learning* adalah menggunakan *Ensemble Learning*. Ide dari teknik ini adalah dengan menggabungkan beberapa algoritma *Machine Learning* atau yang biasa disebut sebagai *base learners*. Penelitian ini menggunakan algoritma atau metode *Logistic Regression* sebagai *base learners* untuk membentuk model *Ensemble Learning*. Tujuan utama dari penelitian ini adalah membandingkan performa (nilai evaluasi dan waktu) dua algoritma *Ensemble Learning* yaitu metode *Bootstrap Aggregating (Bagging)* dan metode *Adaptive Boosting (AdaBoost)*. Penelitian menggunakan sebelas *dataset* dengan klasifikasi *multi-class* yang independen terhadap karakteristik (proporsi data, jumlah data, dan masalah) serta jumlah kelas variabel target berbeda. Hasil penelitian menunjukkan bahwa model yang dibentuk oleh metode *Bagging* cenderung menunjukkan performa nilai yang lebih baik dari metode *AdaBoost* pada metrik evaluasi jika hanya dilihat berdasarkan per *dataset* yang digunakan, dengan rata-rata nilai evaluasi sebesar 72,21% dan 61% untuk *Bagging* serta 66,25% dan 53,7% untuk *AdaBoost* dengan metrik akurasi dan F1 secara berurutan dalam pembentukan model. Namun, hasil uji rata-rata untuk kedua metode menjelaskan bahwa tidak ada metode yang lebih baik dari metode lainnya atau dapat dikatakan bahwa performa metode *Bagging* sama dengan performa metode *AdaBoost*. Hasil uji rata-rata untuk waktu pembentukan kedua model juga memberikan hasil yang tidak signifikan atau dapat dikatakan bahwa lama waktu komputasi untuk membentuk model *Bagging* dan model *AdaBoost* sama.

1 Latar Belakang

Seiring dalam perkembangan zaman, teknologi pun juga mengalami kemajuan yang tidak dapat dihindari. Segala hal yang saat ini dinikmati dan membantu banyak orang di dunia tidak lain adalah hasil dari perkembangan teknologi yang selalu bergerak maju. Salah satu dari sekian banyak ilmu yang mempunyai peranan penting dalam kemajuan teknologi adalah ilmu matematika dan statistika. Ketika ilmu-ilmu ini dipakai dalam dunia nyata seringkali ilmu-ilmu ini berhubungan dengan data. Data pada kenyataannya banyak membantu kehidupan manusia dalam berbagai sektor seperti kesehatan, ekonomi, pendidikan, dan tentunya teknologi. Berbagai data yang terkumpul kemudian akan dipahami dan dianalisis sehingga menghasilkan sesuatu yang dapat bermanfaat dan praktikal. Seiring berjalannya waktu orang berlomba-lomba untuk mencitakan metode yang terbaik untuk dapat mengolah data menjadi semakin efisien dan akurat, hal ini lah yang membuat ada sangat banyak metode pengolahan data [1]. Contoh yang dapat dilihat adalah bagaimana mendeteksi orang yang depresi, mengatasi masalah penyebaran *Coronavirus Disease* (COVID-19), dan masih banyak lagi [2].

Metode yang digunakan untuk menganalisis data-data yang telah diperoleh dikenal sebagai *Machine Learning* yang kemudian tugasnya adalah membantu manusia dalam menemukan pola-pola dari data yang sangat kompleks sehingga bermanfaat. Tujuan akhir yang harus dapat diselesaikan oleh *Machine Learning* adalah untuk membuat model yang baik untuk menjawab sebuah masalah. Salah satu perluasan *Machine Learning* dikenal sebagai *Ensemble Learning*. Metode *Ensemble* melatih beberapa *learners* untuk memecahkan masalah yang sama, proses singkatnya adalah dengan membangun sebuah gabungan dari *learners* dan menyatukan semuanya [3].

Netflix pernah membuat kompetisi untuk menyelesaikan masalah *Supervised Learning* dengan hadiah sebesar USD1.000.000 bagi yang dapat meningkatkan performa *Netflix classifier* sebanyak 10%. Setelah tiga bulan berjalan didapati ada beberapa tim di puncak klasemen yang berhasil meningkatkan performa sebanyak 5%, sampai pada akhirnya ada tim terbaik yang berhasil meningkatkan performa sebesar 8.5% dan hasil akhir menyatakan bahwa beberapa tim yang menduduki puncak klasemen memakai metode *Ensemble* untuk meningkatkan performa klasifikasi pada tanggal 21 September 2009 [1].

2 Metodologi

Penelitian ini dimulai dengan pengumpulan data, melakukan *data preprocessing*, melakukan pembentukan model *Ensemble* dengan basis *Logistic Regression*, evaluasi model dengan *Confusion Matrix*, dan pengujian hipotesis dari nilai evaluasi yang diperoleh dari perhitungan pada *Confusion Matrix*.

2.1 Data

Data yang digunakan dalam penelitian ini diambil dari situs *kaggle.com*. Data yang diambil merupakan data-data dari berbagai sektor (medis, keuangan, astronomi, dan lain-lain). Data-data ini kemudian akan dipisah menjadi dua kriteria yakni data pelatihan (*data train*) dan data pengujian (*data test*) dengan proporsi 80% untuk data pelatihan dan 20% untuk data pengujian yang akan dipilih secara acak.

2.2 Data Preprocessing

Proses ini melewati beberapa tahap yaitu *Exploratory Data Analysis*, *Feature Engineering*, dan *Handling Missing Values*. Proses-proses ini melewati tahap seperti menghitung korelasi dan normalisasi/standarisasi data.

2.2.1 Korelasi

Dalam perangkat lunak python, setelan korelasi yang digunakan adalah korelasi pearson. Fungsi korelasi sendiri adalah untuk mengukur asosiasi linier, mencari dan mengetahui apakah ada hubungan antara dua variabel [4]. Variabel pertama dinotasikan sebagai x dan variabel kedua dinotasikan sebagai y . Bentuk umum korelasi pearson dapat dilihat pada Persamaan 1

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 - (y_i - \bar{y})^2}}, \quad (1)$$

dengan x_i adalah data ke- i pada variabel pertama, \bar{x} adalah rata-rata dari data pertama, y_i adalah data ke- i pada variabel kedua, dan \bar{y} adalah rata-rata dari data kedua.

2.2.2 Normalisasi dan Standarisasi Data

Normalisasi dan standarisasi data adalah metode yang paling umum digunakan dalam melakukan *feature scaling*, scaling dipakai agar proporsi setiap variabel sama. Sebagai contoh fitur yang memiliki data dengan *magnitude* lebih besar mempunyai kecenderungan mendapatkan proporsi yang lebih besar dibanding fitur-fitur dengan *magnitude* yang kecil. Normalisasi adalah teknik *scaling* yang menggeser dan menskala nilai-nilai pada data sehingga berada di antara 0 dan 1. Normalisasi dikenal dengan sebutan *min-max scaling* dengan bentuk

$$X_{baru} = \frac{X - X_{min}}{X_{max} - X_{min}}, \quad (2)$$

dengan X adalah nilai yang ingin di normalisasi, X_{max} dan X_{min} adalah nilai maksimum dan minimum pada fitur tersebut [5]. Berbeda dengan normalisasi, teknik standarisasi mengubah nilai μ dan σ suatu data menjadi 0 dan 1, atau dengan kata lain menjadi bentuk distribusi normal standar

$$z = \frac{x - \mu}{\sigma}, \quad (3)$$

dengan

$$\mu = rata - rata = \frac{1}{N} \sum_{i=1}^N x_i, \quad (4)$$

dan

$$\sigma = standardeviasi = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}. \quad (5)$$

2.3 Pembentukan Model

2.3.1 Model Logistic Regression

Data yang akan digunakan merupakan data dengan klasifikasi *multi-class* dengan label angka $0, 1, 2, \dots, n$ yang merepresentasikan kelas pertama, kedua, sampai kelas ke- n . Nilai variabel dependen Y mempunyai j kategori dengan $j = 0, 1, 2, \dots, n$ dengan nilai kemungkinan respon $\pi_0, \pi_1, \pi_2, \dots, \pi_n$ dan nilai $\sum_{j=0}^n \pi_j = 1$. Kemungkinan $Pr(y = j|x) = \pi_j(x)$ sehingga dapat ditulis menjadi bentuk

$$Pr(y = j|x) = \pi_j(x), \quad (6)$$

dengan $\pi_j(x) = Pr(y = j)$ dan $1 - \pi_j(x) = Pr(y \neq j)$ [6].

2.3.2 Model Bagging

Proses kerja *Bagging* akan meliputi beberapa langkah yang akan dijelaskan sebagai berikut.

1. Proses pembentukan *classifier*, proses ini bertujuan untuk menentukan berapa banyak *classifier* (T) yang akan dibentuk dalam proses *Bagging*. Penentuan jumlah *classifier* yang akan dibentuk dapat menggunakan fungsi *n_estimators*.
2. Menentukan proporsi *classifier*, dalam proses *Bagging* akan ditentukan berapa proporsi distribusi data untuk setiap *classifier* yang ada. Dari total n data yang ada di *training data*, akan ditentukan sejumlah m sampel, dengan nilai $m < n$. Penentuan nilai m dapat diatur menggunakan parameter dalam python dalam fungsi *max_samples*.
3. *Random Sampling with Replacement*, metode ini memungkinkan sebuah data dapat terpilih dua kali dalam *classifier* yang sama. Dalam proses ini akan ditentukan berapa jumlah fitur yang akan dipakai untuk setiap *classifier*. Akan digunakan fungsi *max_features* untuk menentukan jumlah fitur yang dapat digunakan dengan setelan awal menggunakan semua fitur yang ada.
4. Pelatihan model, akan dilakukan pembuatan model untuk setiap *classifier* dengan *base model* yang digunakan adalah *logistic regression* sehingga akan terbentuk sejumlah T model yang nantinya akan digunakan untuk memprediksi nilai target (Y).

5. *Aggregating*, dalam proses ini akan dilakukan pemilihan suara terbanyak dari sejumlah T *classifier* yang ada, kemudian hasil dari pemilihan tersebut akan menjadi hasil akhir dari prediksi yang akan digunakan [7].

Ilustrasi penggunaan algoritma *Bagging* dapat dilihat pada Gambar 2.1.

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;
 Base learning algorithm \mathcal{L} ;
 Number of base learners T .

Process:
 1. **for** $t = 1, \dots, T$:
 2. $h_t = \mathcal{L}(D, \mathcal{D}_{bs})$ % \mathcal{D}_{bs} is the bootstrap distribution
 3. **end**

Output: $H(x) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T \mathbb{I}(h_t(x) = y)$

Gambar 2.1: Algoritma *Bagging*

Sumber: *Ensemble Methods: Foundations and Algorithms* [3]

2.3.3 Model *AdaBoost*

Pada metode *Bagging*, setiap *classifier* mempunyai kekuatan suara (*vote*) yang sama satu dengan yang lain. Sedangkan dalam metode *AdaBoost* setiap *classifier* mempunyai hak kekuatan suara (*vote*) yang berbeda dan proses pembuatan tiap *classifier* dilakukan secara *sequential*. Proses kerja *AdaBoost* akan meliputi beberapa langkah yang akan dijelaskan sebagai berikut.

1. Akan diberikan *sample weight*, di mana pada iterasi pertama setiap sampel (observasi) akan diberikan *weight* yang sama bernilai $\frac{1}{total\ sampel}$.
2. Buat model pertama dan akan didapat hasil prediksi dari model tersebut. Anggap model pertama mempunyai sejumlah T misklasifikasi, kemudian akan dihitung $Total\ Error = \frac{T}{N}$, $N = total\ sampel$.
3. Hitung performa model tersebut dengan fungsi,

$$PerformaModel = \frac{1}{2} \ln \left(\frac{1 - Total\ Error}{Total\ Error} \right). \quad (7)$$

Nilai ini kemudian akan menjadi kekuatan suara (*vote*) untuk model tersebut.

4. Setiap sampel kemudian akan diperbarui sehingga sampel yang salah diprediksi oleh model akan diberikan *weight* yang lebih besar (diperbesar) dan sampel yang berhasil diprediksi secara benar akan diberikan *weight* yang lebih kecil (diperkecil). Sampel-sampel yang salah diprediksi akan dihitung dengan $Weight\ Baru = sample\ weight \times e^{PerformaModel}$. Sementara, untuk sampel-sampel yang berhasil diprediksi dengan benar akan dihitung dengan

$Weight\ Baru = sample\ weight \times e^{-PerformaModel}$. Sehingga, sampel-sampel yang salah diprediksi mempunyai kemungkinan lebih besar untuk terpilih dalam *training* model selanjutnya.

5. Normalisasi sampel, setelah didapatkan *weight* yang baru, lakukan normalisasi sampel agar total *weight* yang baru bernilai 1.

$$Normalisasi\ Weight = \frac{x_i}{\sum_{i=1}^N Total\ Weight\ Baru}, \quad (8)$$

dengan x adalah *weight* baru dari sampel(observasi) dan $i = 1, 2, \dots, N$.

6. Ulang langkah 2-5 hingga mendapatkan sejumlah M model yang ditentukan menggunakan fungsi *n_estimator* pada python [7].

Ilustrasi penggunaan algoritma *AdaBoost* dapat dilihat pada Gambar 2.2.

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;
Base learning algorithm \mathfrak{L} ;
Parameter ν .
Number of learning rounds T .

Process:

1. $w_{1,i} = 1/m$ ($\forall i = 1, \dots, m$).
2. $\beta_1 = 0$.
3. **for** $t = 1, \dots, T$:
4. $h_t = \mathfrak{L}(D, w)$; % Train a learner h_t from D under w
5. **if** $\sum_{i=1}^m w_{t,i} y_i h_t(x_i) \leq \beta_t$ **then** $T = t - 1$; **break**; % Check optimality
6. $H_{i,t} = h_t(x_i)$ ($i = 1, \dots, m$); % Fill a column
7. $(w_{t+1}, \beta_{t+1}) = \arg \min_{w, \beta} \beta$
s.t. $\sum_{i=1}^m w_i y_i h_j(x_i) \leq \beta$ ($\forall j \leq t$)
 $\sum_{i=1}^m w_i = 1$
 $w_i \in [0, \frac{1}{m\nu}]$ ($\forall i = 1, \dots, m$)
8. **end**
9. solve α from the dual solution (w_{T+1}, β_{T+1})

Output: $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

Gambar 2.2: Penggambaran Algoritma *LPBoost* untuk Masalah Klasifikasi *Multi-class*
Sumber: *Ensemble Methods: Foundations and Algorithms* [3]

2.4 Confusion Matrix

Confusion matrix adalah salah satu cara untuk mengukur performa masalah klasifikasi dalam *machine learning*. Metode ini dapat diterapkan dalam masalah yang mempunyai prediksi *binary* maupun *multi-class*. *Output* yang dihasilkan dari *confusion matrix* berupa tabel dengan empat kondisi yang ada dalam tabel. Seperti yang ada pada Gambar 2.3 ditunjukkan bahwa ada kondisi:

- *True Positive* (TP) yang berarti prediksi data positif dan hasilnya benar.
- *True Negative* (TN) yang berarti prediksi data negatif dan hasilnya benar.
- *False Positive* (FP) atau kesalahan tipe pertama, yang berarti prediksi data positif dan hasilnya salah.

- *False Negative* (FN) atau kesalahan tipe dua yang berarti prediksi data negatif dan itu salah.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2.3: Tabel *Confusion Matrix*

Sumber: *Confusion Matrix for Your Multi-Class Machine Learning Model* [8]

2.4.1 Akurasi

Akurasi menjelaskan rasio keakuratan model. Akurasi membandingkan semua sampel yang diprediksi dengan benar dengan keseluruhan sampel

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN}. \quad (9)$$

2.4.2 *F1-score*

F1-score mengkombinasikan nilai presisi dan *recall*. Secara matematik, nilai ini adalah *harmonic mean* dari presisi dan *recall*

$$F1 = 2 \times \frac{presisi \times recall}{presisi + recall} = \frac{2TP}{2TP + FP + FN}. \quad (10)$$

2.5 Pengujian Hipotesis

2.5.1 Uji Normalitas

Setelah mendapatkan hasil akurasi, nilai F1, dan waktu pembentukan model. Kemudian, akan dilakukan uji normalitas menggunakan metode Kolmogorov-Smirnov untuk setiap data yang didapatkan untuk memilih metode yang akan dipakai dalam uji hipotesis. Jika data ditemukan berdistribusi normal, maka akan dipakai uji-t, namun jika ada salah satu data yang tidak normal, akan dipilih salah satu metode dari uji non-parametrik [9]. Pengujian normalitas akan dilakukan menggunakan metode Kolmogorov-Smirnov (KS) dan Shapiro-Wilk (SW) yang menggunakan SPSS. Dengan langkah-langkah penggunaan metode Kolmogorov-Smirnov adalah,

1. susun data yang akan diuji dari nilai terkecil hingga terbesar (x_i),
2. tentukan hipotesis dengan H_0 = data berdistribusi normal dan H_1 = data TIDAK berdistribusi normal,
3. tentukan nilai frekuensi kumulatif (F_{kum}) dan kumulatif proporsi ($F_n(x_i)$),
4. hitung nilai normal untuk setiap data (Z),
5. lihat nilai pada tabel Z untuk menentukan luas di bawah kurva normal baku ($F_0(x_i)$),
6. hitung nilai $|F_n(x_i) - F_0(x_i)|$,
7. cari nilai D_{hitung} dengan $D = \sup_x |F_n(x_i) - F_0(x_i)|$,
8. cari nilai D_{tabel} dari tabel Kolmogorov-Smirnov $D_{\alpha,N}$, dengan α adalah nilai signifikansi dan N adalah total data,
9. bandingkan nilai D_{hitung} dengan nilai D_{tabel} , jika nilai $D_{hitung} < D_{tabel}$, berarti H_0 diterima atau tidak dapat ditolak.

Sementara, langkah-langkah untuk penggunaan metode Shapiro-Wilk adalah,

1. tentukan hipotesis dengan H_0 : data berdistribusi normal dan H_1 : data TIDAK berdistribusi normal,
2. tentukan nilai signifikansi α ,
3. susun data yang akan diuji dari nilai terkecil hingga terbesar (x_i),
4. hitung nilai $D = \sum_{i=1}^n (x_i - \bar{x})^2$,
5. hitung nilai statistik uji Shapiro-Wilk $T_3 = \frac{1}{D} [\sum_{i=1}^n a_i (x_{n-i+1} - x_i)]^2$ dengan nilai a_i adalah koefisien *test* Shapiro-Wilk yang didapat dari tabel Shapiro-Wilk,
6. lihat tabel Shapiro-Wilk untuk melihat letak nilai T_3 , jika nilai $T_3 > \alpha$, berarti H_0 diterima atau tidak dapat ditolak.

2.5.2 Uji Kesamaan Varians

Akan dilakukan uji kesamaan varians atau biasa disebut sebagai uji homogenitas untuk menguji kesamaan dua atau lebih populasi yang berbeda. Pertama, akan diuji kesamaan varians antar tiap metode yang dipakai, yaitu *Bagging* dan *AdaBoost* untuk akurasi, nilai F1, dan Kombinasi. Kedua, akan diuji kesamaan varians antar metrik evaluasi, yaitu akurasi, nilai F1, dan kombinasi untuk model *Bagging* dan *AdaBoost*. Hipotesis pada uji kesamaan varians adalah H_0 : tidak ada perbedaan varians antar populasi dan H_1 : setidaknya ada satu populasi yang memiliki varians berbeda. Hasil uji ini kemudian akan dipakai untuk memilih uji yang akan dipakai untuk langkah selanjutnya. Pengujian akan dilakukan menggunakan Minitab.

2.5.3 Uji Selisih Rataan Dua Populasi

Uji selisih rataan dua populasi dikenal sebagai uji-t. Uji selisih rataan dua populasi digunakan untuk menentukan apakah ada bukti bahwa rataan populasi berbeda secara signifikan. Hipotesis pada uji ini adalah $H_0 : \mu_1 - \mu_2 = 0$ (perbedaan rataan dua kelompok sama dengan 0 atau rataan dua kelompok sama) dan $H_1 : \mu_1 - \mu_2 \neq 0$ (perbedaan rataan dua kelompok tidak sama dengan 0 atau rataan dua kelompok tidak sama) atau $H_1 : \mu_1 - \mu_2 > 0$ (perbedaan rataan dua kelompok lebih dari 0 atau kelompok satu memiliki rataan yang lebih besar dari kelompok dua) atau $H_1 : \mu_1 - \mu_2 < 0$ (perbedaan rataan dua kelompok kurang dari 0 atau kelompok dua memiliki rataan yang lebih besar dari kelompok satu).

Asumsi dan kondisi yang harus dipenuhi untuk melakukan uji-t adalah data harus berdistribusi normal, memiliki varians yang sama, dan independen. Oleh karena itu, sebelumnya dilakukan uji normalitas untuk data yang diperoleh. Kemudian, jika data memiliki varians yang berbeda, tetap dapat dilakukan pengujian namun akan ada perbedaan pada bagian derajat kebebasan [10].

Akan diuji selisih rataan untuk empat hal. Pertama, selisih rataan antara performa *Bagging* dan *AdaBoost* dengan metrik akurasi. Kedua, selisih rataan antara performa *Bagging* dan *AdaBoost* dengan metrik nilai F1. Ketiga, selisih rataan antara performa *Bagging* dan *AdaBoost* dengan kombinasi metrik akurasi dan nilai F1 berdasarkan proporsi variabel target. Keempat, selisih rataan antara waktu pembentukan model *Bagging* dan *AdaBoost*. Hasil dari uji selisih rataan ini kemudian akan dipakai untuk menarik kesimpulan secara statistik terhadap data yang diperoleh dari hasil pembentukan dan pengolahan model. Pengujian akan dilakukan menggunakan Minitab.

2.5.4 Uji Rataan Untuk Lebih Dari Dua Populasi

Metrik yang digunakan untuk mengukur performa metode *Bagging* dan *AdaBoost* ada dua, yaitu akurasi dan nilai F1. Namun, akan diberikan perlakuan tambahan pada metrik pengukuran kedua metode. *Dataset* yang seimbang akan diukur menggunakan akurasi, sedangkan *dataset* yang tidak seimbang akan diukur menggunakan nilai F1. Kemudian akan dibuat kelompok baru yang mengombinasikan kedua metrik berdasarkan proporsi *dataset*. Kemudian, akan diuji rataan antara metrik akurasi, nilai F1, dan kombinasi untuk metode *Bagging*, serta rataan antara metrik akurasi, nilai F1, dan kombinasi untuk metode *Boosting*. Uji ini dilakukan untuk melihat apakah proporsi sebuah data menentukan perbedaan antara metrik pengukuran yang dipakai.

Uji rataan untuk lebih dari dua populasi atau yang biasa dikenal sebagai *Analysis of Variance* (ANOVA) adalah metode yang dikembangkan dari uji satu rataan dan uji dua rataan. ANOVA dapat melakukan pengujian rataan untuk lebih dari dua populasi atau kelompok. Hipotesis pada ANOVA adalah H_0 : tidak ada perbedaan rataan antar populasi dan H_1 : setidaknya ada satu populasi yang memiliki rataan berbeda.

Asumsi dan kondisi yang harus dipenuhi untuk melakukan uji ANOVA adalah

data harus berdistribusi normal, mempunyai varians yang sama, dan independen. Oleh karena itu, ada uji normalitas dan uji kesamaan varians (homogenitas) untuk mengetahui apakah data yang diperoleh sudah memenuhi asumsi. Pengujian akan dilakukan menggunakan Minitab [11].

3 Hasil dan Pembahasan

Penjelasan hasil pengolahan data dimulai dengan pencatatan hasil olah data yang diperoleh menggunakan Python yang kemudian dievaluasi menggunakan *Confusion Matrix* dan hasil pengujian hipotesis.

3.1 Hasil Pencatatan

Setelah melakukan pengolahan terhadap data-data, diperoleh hasil pencatatan untuk akurasi, nilai F1, dan waktu yang dibutuhkan untuk pembentukan model yang akan ditunjukkan pada Tabel 3.1.

Tabel 3.1: Hasil Pencatatan

Dataset	Proporsi	Akurasi			F1-score			Kombinasi			Waktu (detik)	
		<i>Bagging</i>	<i>AdaBoost</i>		<i>Bagging</i>	<i>AdaBoost</i>		<i>Bagging</i>	<i>AdaBoost</i>		<i>Bagging</i>	<i>AdaBoost</i>
Glass	<i>Imbalance</i>	0,674418605	0,581395349		0,622804659	0,42037037		0,622804659	0,42037037		0,767	0,309
Sky	<i>Imbalance</i>	0,9745	0,9555		0,97169818	0,948574543		0,948574543	0,948574543		1,63	12,90
Heart	<i>Imbalance</i>	0,576086957	0,586956522		0,327887808	0,316894289		0,327887808	0,316894289		1,82	0,575
Customer	<i>Balance</i>	0,49784792	0,481348637		0,456284334	0,481348637		0,49784792	0,481348637		2,27	3,09
Handphone	<i>Balance</i>	0,926829268	0,926829268		0,926829268	0,926966315		0,926829268	0,926829268		3,29	6,15
Car	<i>Imbalance</i>	0,930635838	0,861271676		0,857892062	0,776883425		0,857892062	0,776883425		4,27	12,10
Wine	<i>Imbalance</i>	0,6	0,58125		0,29303082	0,286192141		0,29303082	0,286192141		4,70	3,75
BMI	<i>Balance</i>	0,885167464	0,600478469		0,885167464	0,580968389		0,885167464	0,600478469		6,34	2,24
Muscle	<i>Balance</i>	0,341609589	0,290667808		0,344351878	0,292972419		0,41609589	0,290667808		7,07	4,29
Tech Student	<i>Balance</i>	0,6765	0,65375		0,675105649	0,653903339		0,6765	0,65375		11,30	4,90
Diamond	<i>Imbalance</i>	0,86	0,768333333		0,349390386	0,222741477		0,349390386	0,222741477		7,97	9,64
Rata-rata		0,722145058	0,662525551		0,610040228	0,537074122					4,675	5,449

Jika dilihat berdasarkan metrik evaluasi untuk setiap saja dapat dilihat sekilas bahwa model *Bagging* cenderung menghasilkan nilai yang lebih baik dari pada model *AdaBoost*. Hal ini sama seperti hasil penelitian dari salah satu tinjauan pustaka, di mana metode *Bagging* memberikan hasil yang lebih baik dibandingkan dengan model *ensemble* lainnya. Kemudian akan dilakukan uji normalitas untuk setiap hasil pengolahan data yang didapat. Hasilnya, kemudian akan digunakan untuk menentukan metode pengujian hipotesis yang akan digunakan [12].

3.2 Hasil Pengujian Hipotesis

Setelah melakukan serangkaian pengujian dan pencatatan, akan dilihat rangkuman hasil pengujian yang telah dilakukan. Rangkuman yang akan diperlihatkan adalah hasil uji normalitas, uji kesamaan varians, uji selisih rata-rata dua populasi, dan uji ANOVA. Berikut adalah rangkuman hasil pengujian yang dilakukan.

- Rangkuman hasil uji normalitas dapat dilihat pada Tabel 3.2.

Tabel 3.2: Rangkuman Hasil Uji Normalitas

		<i>p-value</i>	
		Kolmogorov-Smirnov	Shapiro-Wilk
Akurasi	<i>Bagging</i>	0,2	0,364
	<i>AdaBoost</i>	0,2	0,569
Nilai F1	<i>Bagging</i>	0,2	0,079
	<i>AdaBoost</i>	0,2	0,253
Kombinasi	<i>Bagging</i>	0,2	0,075
	<i>AdaBoost</i>	0,2	0,254
Waktu	<i>Bagging</i>	0,2	0,498
	<i>AdaBoost</i>	0,2	0,218

- Rangkuman hasil uji kesamaan varians dapat dilihat pada Tabel 3.3.

Tabel 3.3: Rangkuman Hasil Uji Kesamaan Varians

Uji Kesamaan Varians	
	<i>p-value</i>
Akurasi <i>Bagging</i> dengan Akurasi <i>AdaBoost</i>	0,911
Nilai F1 <i>Bagging</i> dengan Nilai F1 <i>AdaBoost</i>	0,940
Kombinasi <i>Bagging</i> dengan Kombinasi <i>AdaBoost</i>	0,991
Waktu <i>Bagging</i> dengan Waktu <i>AdaBoost</i>	0,362
Akurasi, Nilai F1, dan Kombinasi <i>Bagging</i>	0,697
Akurasi, Nilai F1, dan Kombinasi <i>AdaBoost</i>	0,654

- Rangkuman hasil uji selisih rata-rata dua populasi dapat dilihat pada Tabel 3.4.

Tabel 3.4: Rangkuman Hasil Uji Selisih Rataan Dua Populasi

Uji Selisih Rataan Dua Populasi	
	<i>p-value</i>
Akurasi Bagging dengan Akurasi AdaBoost	0,250
Nilai F1 Bagging dengan Nilai F1 AdaBoost	0,262
Kombinasi Bagging dengan Kombinasi AdaBoost	0,261
Waktu Bagging dengan Waktu AdaBoost	0,680

- Rangkuman hasil uji ANOVA dapat dilihat pada Tabel 3.5.

Tabel 3.5: Rangkuman Hasil Uji ANOVA

Uji ANOVA	
	<i>p-value</i>
Akurasi, Nilai F1, dan Kombinasi Bagging	0,484
Akurasi, Nilai F1, dan Kombinasi AdaBoost	0,391

4 Kesimpulan

Pada penelitian ini telah dilakukan pembentukan dan perbandingan performa dua metode *ensemble* yaitu *Bagging* dan *AdaBoost*. Penelitian dilakukan menggunakan sebelas *dataset* dengan variabel target *multi-class* dengan terlebih dahulu melakukan pembersihan data, kemudian menentukan *base learners* yaitu *logistic regression*, dan terakhir melakukan pembentukan model *Bagging* dan *AdaBoost*. Hasil prediksi dari kedua model diukur berdasarkan metrik evaluasi nilai akurasi, nilai F1, dan perbandingan lama waktu pembuatan kedua model.

Rata-rata hasil yang diperoleh dari sebelas *dataset* untuk nilai akurasi, nilai F1, dan waktu pemrosesan, secara berurutan, adalah 72,21%, 61%, dan 4,675 detik untuk metode *Bagging*, serta 66,25%, 53,7%, dan 5,449 detik untuk metode *AdaBoost*. Uji hipotesis dilakukan pada kesebelas hasil percobaan untuk mengetahui apakah terdapat perbedaan yang signifikan antara performa kedua model. Hasil yang telah didapat berdasarkan proses pengolahan dan analisis data dapat disimpulkan menjadi poin-poin berikut.

1. Performa model *Bagging* berdasarkan metrik-metrik evaluasi cenderung memberikan nilai yang lebih baik dibandingkan model *AdaBoost* jika hanya dilihat per *dataset*. Namun, jika dilihat berdasarkan pengujian hipotesis, kedua metode tidak memiliki perbedaan yang signifikan secara statistik, sehingga tidak ada metode yang lebih baik dari metode lainnya.
2. Waktu pembentukan model *Bagging* dan *AdaBoost* tidak menunjukkan perbedaan yang jelas jika dilihat per *dataset*. Hal ini juga didukung dengan hasil pengujian hipotesis yang menyatakan tidak ada perbedaan waktu pembentukan model yang signifikan antara model *Bagging* dan *AdaBoost*.

3. Berdasarkan uji hipotesis, pemilihan metrik dalam mengevaluasi *dataset* dengan proporsi kelas yang seimbang maupun tidak seimbang tidak memengaruhi hasil perbandingan yang dilakukan.

Pustaka

- [1] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2):75–79, 2007.
- [2] Haihua Jiang, Bin Hu, Zhenyu Liu, Gang Wang, Lan Zhang, Xiaoyu Li, and Huanyu Kang. Detecting depression using an ensemble logistic regression model based on multiple speech features. *Computational and Mathematical Methods in Medicine*, 2018:1–9, 2018.
- [3] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2019.
- [4] Wilhelm Kirch. Pearson’s correlation coefficient. *Encyclopedia of Public Health*, pages 1090–1091, 2008.
- [5] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
- [6] William Mendenhall, Terry Sincich, and Nancy S Boudreau. *A second course in statistics: regression analysis*, volume 5. Prentice Hall Upper Saddle River, NJ, 1996.
- [7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [8] Joydwip Mohajon. Confusion matrix for your multi-class machine learning model. *Towards Data Science*, 2020.
- [9] Atilla Halil ELHAN and Ersöz TÜCCAR. Investigation of four different normality tests in terms of type 1 error rate and power under different distributions. *Turk J Med Sci*, 36(3):171–176, 2006.
- [10] Tae Kyun Kim. T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6):540, 2015.
- [11] Susan Troncoso Skidmore and Bruce Thompson. Bias and precision of some classical anova effect sizes when assumptions are violated. *Behavior research methods*, 45(2):536–546, 2013.
- [12] Yanli Wu, Yutian Ke, Zhuo Chen, Shouyun Liang, Hongliang Zhao, and Hao-yuan Hong. Application of alternating decision tree with adaboost and bagging ensembles for landslide susceptibility mapping. *Catena*, 187:104396, 2020.