Penghitungan k-NN pada Adaptive Synthetic-Nominal (ADASYN-N) dan Adaptive Synthetic-kNN (ADASYN-kNN) untuk Data Nominal-Multi Kategori

Abstrak

Pada penelitian ini disajikan tentang contoh proses penghitungan k-NN pada teknik *oversampling Adaptive Synthetic-Nominal* (ADASYN-N) dan *Adaptive Synthetic*-kNN (ADASYN-kNN) untuk mengatasi masalah ketidakseimbangan (*imbalanced*) kelas pada dataset dengan fitur *nominal-multi categorie*s. Percobaan penghitungan k-NN menggunakan contoh dataset yang memiliki 10 instances dengan 4 fitur, yang mana masing-masing fiturnya memiliki 3 kategori (*multi-categories*). Contoh dataset untuk percobaan penghitungan tersebut terdistribusi ke dalam 2 kelas, yaitu kelas A terdapat 3 *instance*s dan kelas B dengan 7 *instance*s. Selanjutnya hasil penghitungan k-NN tersebut diujikan pada sebuah dataset dengan fitur *nominal-multi categorie*s yang memiliki distribusi kelas yang tidak seimbang. Kemudian dataset di-*oversampling* dengan metode ADASYN-N dan ADASYN-kNN, kemudian dilakukan uji klasifikasi menggunakan metode *Random Forests.* Hasil klasifikasi dibandingkan akurasinya antara dataset asli dan dataset dengan teknik *oversampling* ADASYN-N serta ADASYN-kNN dan menunjukkan bahwa teknik oversampling ADASYN-N dapat meningkatkan akurasi klasifikasi sebanyak 9,05% dari dataset asli, sedangkan ADASYN-kNN meningkatkan akurasi klasifikasi sebanyak 7,84% dari dataset asli.

Keywords: *penghitungan k-NN; ADASYN; imbalanced data; nominal; k-NN; multi categories*

1 Pendahuluan

Banyak permasalahan data mining melibatkan *imbalanced data* (ketidakseimbangan data). Dataset dengan ketidakseimbangan kelas ini terjadi karena rasio yang tidak seimbang antara kasus yang satu dengan kasus yang lainnya. Ketidakseimbangan kelas ini akan merugikan pada penelitian bidang *data mining* karena *machine learning* pada *data mining* memiliki kesulitan dalam mengklasifikasikan kelas minoritas (jumlah instance yang kecil) dengan benar. Beberapa algoritme mengasumsikan bahwa distribusi kelas yang diuji adalah seimbang sehingga dalam beberapa kasus menjadikan kesalahan dalam mengklasifikasikan hasil pada tiap kelas.

Terdapat beberapa pendekatan untuk penanganan ketidakseimbangan, salah satunya dengan menggunakan metode sampling data asli baik pada kelas mayoritas (*under-sampling*) maupun kelas minoritas (*over-sampling*). *Under-sampling* merupakan metode untuk menyeimbangkan kelas dengan cara mengurangi instance pada kelas mayoritas secara acak. Namun, pada metode *under-sampling* memiliki resiko hilangnya informasi dan data yang dianggap penting untuk proses pengambilan keputusan oleh *machine learning*.

Sedangkan o*ver-sampling* merupakan metode penyeimbangan distribusi kelas dengan mereplikasi instance pada kelas minoritas secara acak. *Over-sampling* meningkatkan kemungkinan munculnya *overfitting* karena menduplikasi *instance* secara sama persis. He, dkk mengajukan metode untuk pendekatan *sampling* pada pembelajaran dengan dataset tidak seimbang dengan fitur numerik yaitu ADASYN [1]. Ide utama dari ADASYN adalah menggunakan bobot distribusi untuk data pada kelas minoritas berdasarkan pada tingkat kesulitan belajar, dimana data sintesis dihasilkan dari kelas minoritas yang susah untuk belajar dibandingkan dengan data minoritas yang lebih mudah untuk belajar.

Untuk penanganan data dengan fitur nominal, Kurniawati [2] mengembangkan ADASYN-N dan ADASYN-KNN yang merupakan pengembangan dari metode ADASYN. ADASYN-N dan ADASYN-KNN ini disebut dapat menangani ketidakseimbangan data dengan fitur nominal dengan jumlah kategori pada masing-masing fiturnya adalah 2. Akan tetapi, ADASYN-N maupun ADASYN-KNN baru diuji pada satu dataset dengan uji klasifikasi menggunakan metode Naïve Bayes Classifier. Kedua metode tersebut kemudian dibandingkan dengan SMOTE-N[3] dan menunjukkan bahwa ADASYN-N dapat meningkatkan akurasi lebih baik dari SMOTE-N sedangkan ADASYN-KNN menunjukkan performa akurasi dari kedua metode tersebut.

Karena pada penelitian oleh Kurniawati tersebut penghitungan kNN dilakukan pada sebuah dataset dengan 2 kategori pada masing-masing fiturnya, maka pada penelitian ini, dipaparkan penghitungan kNN pada dataset dengan lebih dari dua kategori nominal. Kemudian dataset di-*oversampling* dengan metode ADASYN-N dan ADASYN-kNN, kemudian dilakukan uji klasifikasi menggunakan metode *Random Forests.* Hasil klasifikasi dibandingkan akurasinya antara dataset asli dan dataset dengan teknik *oversampling* ADASYN-N serta ADASYN-kNN.

2 Metodologi

2.1 Penghitungan kNN

Untuk menghitung kNN setiap data, perlu dilakukan penghitungan menggunakan persamaan *euclidean distance* sebagai berikut:

$$D(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \quad (1)$$

Dengan kasus fitur nominal multi categories, maka rumusan euclidean distance menjadi:

$$D(D_1, D_2) = \sqrt{\sum_{k=1}^{n}\left(D_{1,k} - D_{2,k}\right)^2} \quad (2)$$

dengan$\mathrm{D}_1 D_1$ dan$\mathrm{D}_2 2$ adalah data yang diukur jarak *euclidean*-nya, *k* adalah fitur yang terdapat pada data. Pada kasus *nominal multi categorie*s penghitungan $\mathrm{D}_{1,\mathrm{k}}$-$\mathrm{D}_{2,\mathrm{k}}$ $D_{1,k} - D_{2,k}$ menggunakan persamaan $\delta\left(\mathrm{F}_{\mathrm{i,Ca}}, \mathrm{F}_{\mathrm{i,Cb}}\right)$ $\delta\left(F_{i,Ca}, F_{i,Cb}\right)$ di mana $F_{i,Ca}$ adalah fitur ke-*i* dengan kategori *a*. Selanjutnya, menghitung *distanc*e tiap fitur menggunakan persamaan:

$$\delta\left(F_{i,Ca}, F_{i,Cb}\right) = \sum_{i=1}^{n}\left|\frac{c_{1i}}{c_1} - \frac{c_{2i}}{c_2}\right|^{\mathrm{k}} \quad (3)$$

3 ADASYN-Nominal (ADASYN-N) dan Adaptive Synthetic – kNN (ADASYN-kNN)

ADASYN merupakan metode untuk pendekatan s*ampling* pada pembelajaran dengan dataset yang tidak seimbang yang diajukan oleh He, dkk[1]. Ide utama dari ADASYN adalah menggunakan bobot distribusi untuk data pada kelas minoritas berdasarkan pada tingkat kesulitan belajar, sehingga data sintesis dihasilkan dari kelas minoritas yang susah untuk

belajar dibandingkan dengan data minoritas yang lebih mudah untuk belajar. ADASYN meningkatkan pembelajaran dengan dua cara. Pertama, mengurangi bias yang diakibatkan oleh ketidakseimbangan kelas dan yang kedua secara adaptif menggeser batas keputusan klasifikasi terhadap kesulitan data.

ADASYN-N merupakan pengembangan dari ADASYN yang diajukan oleh Kurniawati [3] dengan pendekatan data dengan tipe nominal. *Nearest neighbor* pada ADASYN-N dihitung menggunakan versi modifikasi dari Value Difference Metric (VDM) seperti pada SMOTE-N yang diajukan oleh Chawla, dkk [4]. VDM melihat pada nilai fitur yang overlap terhadap semua vektor fitur. Matriks mendefinisikan jarak antara nilai fitur yang sesuai untuk vektor fitur yang dibuat. Jarak δ antara dua nilai fitur yang sesuai didefinisikan seperti pada Persamaan (3). Berikut prosedur dari metode ADASYN-N:

Input

(1) Training dataset D_{tr} dengan *m* sampel $\{x_i, y_i\}, i = 1, \dots, m$ dimana x_i adalah *instance* dalam n dimensional *feature space* X dan $y_s \in Y = \{1, \dots, C\}$ adalah label identitas kelas dengan jumlah *instance* terbanyak. Tentukan $m_s$dan $m_l$sebagai jumlah *instance* kelas minoritas dan jumlah *instance* kelas mayoritas. Oleh karena itu, $m_{sc} \leq m_l$ dan $\sum m_{sc} + m_l = m$.

Prosedur

(1) Kalkulasi derajat ketidakseimbangan kelas:

$$d_c = m_{sc}/m_l \tag{4}$$

(2) Jika $d_c < d_{th}$ (d_{th} adalah penetapan *threshold* untuk derajat tolerasi maksimum dari rasio ketidakseimbangan kelas):

(a) Hitung jumlah *instance* data sintesis yang perlu digeneralisasi untuk kelas minoritas ke-*c*:

$$G_c = (m_l - m_{sc}) \times \beta \tag{5}$$

Di mana $\beta \in [0,1]$ adalah parameter yang digunakan untuk menetapkan level balance yang diinginkan setelah generalisasi data sintesis. $\beta = 1$berarti data yang sepenuhnya seimbang dibuat setelah proses generalisasi.

(b) Untuk setiap instance $x_i \in$ minority class, temukan *k-nearest neighbors* berdasarkan *Euclidean distance* pada n *dimensional space*, dan kalkulasi rasio r_i yang didefinisikan sebagai:

$$r_i = \frac{\Delta_i}{K}, i = 1, \dots, m_{sc} \tag{6}$$

Di mana $\Delta_i$adalah jumlah *instance* pada *nearest neighbor* yang termasuk kelas y_s (mayoritas) atau termasuk semua kelas kecuali y_{kc} (minoritas), oleh karena itu $x_i \in [0,1]$, y_{kc} adalah kelas yang dievaluasi.

(c) Normalisasi r_i, sehingga $\hat{r}_i$ adalah distribusi kerapatan (*density distribution*) ($\sum_i \hat{r}_i = 1$)

$$\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i \quad (7)$$

(d) Hitung jumlah dari *instance* data sintesis yang perlu dihasilkan pada setiap *instance* minoritas x_i:

$$g_i = \hat{r}_i \times G_c \quad (8)$$

Dimana G_c adalah total jumlah dari instance data sintesis yang perlu untuk dihasilkan untuk kelas minoritas ke-c yang dijelaskan pada Persamaan (5).

(e) Untuk setiap *instance* data kelas minoritas x_i, generasi *instance* data sintesis sebanyak g_i.

Sedangkan ADASYN-KNN merupakan pengembangan dari ADASYN-N dengan pengembangan pada Prosedur (2e) atau prosedur untuk menghasilkan instance data sintesis sebanyak g_i. Pada ADASYN-KNN, data sintesis dihasilkan dari nearest neighbor instance yang dievaluasi. Atribut sintesis dihasilkan dengan melakukan voting berdasarkan pada kemunculan atribut dari nearest neighbor. Kemudian, instance sintesis yang dihasilkan diduplikasi sebanyak g_i.

Prosedur

(1) Kalkulasi derajat ketidakseimbangan kelas: persamaan (4)

(2) Jika $d_c < d_{th}$ then (d_{th} adalah penetapan *threshold* untuk derajat tolerasi maksimum dari rasio imbalance class): (Prosedur 2a sampai 2d sama dengan ADASYN-N)

Untuk setiap instance data kelas minoritas x_i, generate instance data sintesis berdasarkan pada langkah berikut:

a. Cari *nearest neighbor* dari *instance* data kelas minoritas x_i.
b. Lakukan majority voting untuk setiap atribut pada *instance nearest neighbor*.
c. Hasilkan *instance* baru dengan atribut berdasarkan pada *majority voting*.
d. Duplikasi *instance* baru sebanyak g_i.

4 Random Forest

Random Forest pertama kali dikenalkan oleh Breiman [5]. Dalam penelitiannya menunjukkan kelebihan *random forest* antara lain dapat menghasilkan error yang lebih rendah, memberikan hasil yang bagus dalam klasifikasi, dapat mengatasi data training dalam jumlah sangat besar secara efisien, dan metode yang efektif untuk mengestimasi missing data.

Random Forest termasuk metode ansambel yang mutakhir [6]. *Random Forest* melakukan pengacakan bukan pada data latih, melainkan ke dalam algoritma pembelajaran dasar . Secara khusus, *Random Forest* melatih keputusan acak sebagai pembelajar dasar dengan pemilihan fitur acak. Saat membuat pohon keputusan komponen, pada setiap langkah pembagian seleksi, *Random Forest* terlebih dahulu memilih subset fitur secara acak, dan

kemudian melakukan prosedur pemilihan pemisahan konvesional di dalam subset fitur yang dipilih.

5 Jalannya Penelitian

5.1 Penghitungan k-NN

Dimisalkan dataset dengan 10 data yang terbagi ke dalam 2 kelas, memiliki 4 fitur yang masing-masing terdapat 3 kategori seperti yang ditunjukkan pada Tabel 1.

Tabel 1 Contoh dataset

No	F1	F2	F3	F4	*Class*
1	C1	C1	C1	C1	A
2	C1	C2	C1	C2	A
3	C2	C2	C1	C3	A
4	C2	C2	C1	C1	B
5	C2	C3	C3	C2	B
6	C2	C1	C2	C3	B
7	C2	C3	C2	C1	B
8	C2	C3	C3	C2	B
9	C3	C2	C3	C3	B
10	C3	C3	C1	C1	B

Dari data tersebut, kelas mayoritas (y_l) = B berjumlah 7 *instance*s, sedangkan kelas minoritas (y_{sc}) = A berjumlah 3 *instance*s. Selanjutnya dihitung kemunculan kategori pada masing-masing fitur yang ditunjukkan pada Tabel 2.

Tabel 2 Kemunculan kategori masing-masing fitur

Fitur	A_{c1}	B_{c1}	Total	A_{c2}	B_{c2}	Total	A_{c3}	B_{c3}	Total
F1	2	0	**2**	1	5	**6**	0	2	**2**
F2	1	1	**2**	2	2	**4**	0	4	**4**
F3	3	2	**5**	0	2	**2**	0	3	**3**
F4	1	3	**4**	1	2	**3**	1	2	**3**

Untuk menghitung distance antar kategori masing-masing fitur digunakan persamaan (3), sebagai contoh untuk fitur F1:

$$\delta(F_{1,c1}, F_{1,c2}) = \left|\frac{A_{F1,c1}}{Total_{F1,c1}} - \frac{A_{F1,c2}}{Total_{F1,c2}}\right| + \left|\frac{B_{F1,c1}}{Total_{F1,c1}} - \frac{B_{F1,c2}}{Total_{F1,c2}}\right|$$

$$= \left|\frac{2}{2} - \frac{1}{6}\right| + \left|\frac{0}{2} - \frac{5}{6}\right|$$

$$= 1{,}667$$

$$\delta(F_{1,c2}, F_{1,c3}) = \left|\frac{A_{F1,c2}}{Total_{F1,c2}} - \frac{A_{F1,c3}}{Total_{F1,c3}}\right| + \left|\frac{B_{F1,c2}}{Total_{F1,c2}} - \frac{B_{F1,c3}}{Total_{F1,c3}}\right|$$

$$= \left|\frac{1}{6} - \frac{0}{2}\right| + \left|\frac{5}{6} - \frac{2}{2}\right|$$

$$= 0{,}333$$

$$\delta(F_{1,c1}, F_{1,c3}) = \left|\frac{A_{F1,c1}}{Total_{F1,c1}} - \frac{A_{F1,c3}}{Total_{F1,c3}}\right| + \left|\frac{B_{F1,c1}}{Total_{F1,c3}} - \frac{B_{F1,c3}}{Total_{F1,c3}}\right|$$

$$= \left|\frac{2}{2} - \frac{0}{2}\right| + \left|\frac{0}{2} - \frac{2}{2}\right|$$

$$= 2$$

Sehingga untuk keseluruhan fitur jarak antar kategori masing-masing fitur ditunjukkan pada Tabel 3:

Tabel 3 Jarak antar kategori pada masing-masing fitur

Fitur	Distance c1-c2	Distance c2-c3	Distance c1-c3
F1	1,667	2	0,333
F2	0	1	1
F3	1,2	0	1,2
F4	0,167	0	0,167

Kemudian dihitung jarak euclidean antara data 1 dengn data lainnya dengan Persamaan (2). Sebagai contoh penghitungan jarak euclidean antara data 1 dan data 9 sebagai berikut:

$$\mathrm{D}(D_1, D_9) = \sqrt{\sum_{k=1}^{4}(D_{1,k} - D_{9,k})^2}$$

$$\mathrm{D}(D_1, D_9) = \sqrt{\begin{matrix}(D_{1,1} - D_{9,1})^2 + (D_{1,2} - D_{9,2})^2 + \\ (D_{3,3} - D_{9,3})^2 + (D_{1,4} - D_{9,4})^2\end{matrix}}$$

$$\mathrm{D}(D_1, D_9) = \sqrt{\begin{matrix}(D_{1,1=c1} - D_{9,1=c3})^2 + (D_{1,2=c1} - D_{9,2=c2})^2 + \\ (D_{1,3=c1} - D_{9,3=c3})^2 + (D_{1,4=c1} - D_{9,4=c3})^2\end{matrix}}$$

Bila kategori fitur ke-k pada D_1dan D_9 sama, maka jarak fitur = 0, sedangkan bila kategori yang muncul berbeda maka jarak fitur dihitung sehingga,

$$\mathrm{D}(D_1, D_9) = \sqrt{(0{,}167)^2 + (0)^2 + (1{,}2)^2 + (0{,}167)^2}$$

$$\mathrm{D}(D_1, D_9) = \sqrt{1{,}495778}$$

$$\mathrm{D}(D_1, D_9) = 1{,}22302$$

6 Penghitungan ADASYN

Untuk contoh penghitungan ADASYN dari dataset pada Tabel 1, maka dilakukan prosedur sebagai berikut:

(1) Kalkulasi derajat ketidakseimbangan kelas dengan Persamaan (4), sehingga:

$$d_c = \frac{3}{7} = 0{,}428$$

(2) Dengan $\mathrm{d_{th}} = 0{,}75$, hasil kalkulasi ketidakseimbangan kelas di atas memenuhi kondisi $\mathrm{d_c} < \mathrm{d_{th}}$, maka dilanjutkan ke prosedur penghitungan selanjutnya:

(a) Menghitung jumlah instance data sintesis yang perlu di-*generate* menggunakan Persamaan (5) dengan $\beta = 0{,}9$, maka:

$$G_c = (7-3)x0{,}9 = 3{,}6$$

(b) Menghitung rasio ri menggunakan Persamaan (6). Sebelumnya dilakukan penghitungan Δ untuk setiap data pada kelas minoritas A:

- Evaluasi nearest neighbor pada data pertama (D1) dengan nilai K = 5 yang ditunjukkan pada Tabel IV.

Tabel 4 Nearest Neighbor untuk Data 1 (D1)

Data	*Kelas*	*Jarak*
D2	A	0,167
D4	B	1,667
D3	A	1,675
D6	B	2,0605
D10	B	2,236

Tabel 4 menunjukkan terdapat 3 data dengan kelas selain kelas A, sehingga $\Delta_{\mathrm{D1}} = 3$. Maka, rasio untuk data D1 adalah

$$r_{D1} = \frac{\Delta_{D1}}{K}$$
$$r_{D1} = \frac{3}{5}$$
$$r_{D1} = 0{,}6$$

- Dengan cara yang sama maka diperoleh nilai Δ untuk setiap data seperti pada Tabel 5.

Tabel 5 NN untuk setiap data di kelas A

Data Evaluasi	Data terdekat	Δ_i	r_i
D1	D2,D4,D3,D6,D10	3	0,6
D2	D1, D3, D4, D6, D10	3	0,6
D3	D4, D10, D6, D9, D5/D8	5	1

(c) Normalisasi r_i untuk mendapatkan *density distribution* ($\hat{r}$) dengan Persamaan (7) sehingga didapatkan hasil pada Tabel 6.

Tabel 6 *Density Distribution* untuk data kelas a

Data Evaluasi	$\hat{r}_i$

D1	0,2727
D2	0,2727
D3	0,4545

(d) Menghitung jumlah duplikasi instance sintesis untuk tiap data ke-i (x_i) dengan Persamaan (8). Sebagai contoh data ke-1 maka,

$$g_1 = \hat{r}_i x G_c$$

$$g_1 = 0{,}2727 x 3{,}6$$

$$g_1 = 0{,}982$$

Untuk semua data diperoleh $g_i$dan dilakukan pembulatan untuk mendapatkan jumlah duplikasi data sintesis nya seperti ditunjukkan Tabel 7.

Tabel 7 jumlah duplikasi data sintesis

Data	$\hat{r}_i$	g_i	sintesis
D1	0,2727	0,982	1
D2	0,2727	0,982	1
D3	0,4545	1,6362	2

(e) Untuk ADASYN-N, replikasi dilakukan secara langsung sejumlah nilai sintesis yang didapat dan ditambahkan pada dataset awal sehingga menghasilkan dataset baru seperti pada Tabel 8.

Tabel 8 Dataset baru setelah penambahan data sintesis dengan metode ADASYN-N

No	F1	F2	F3	F4	class
1	C1	C1	C1	C1	A
2	C1	C2	C1	C2	A
3	C2	C2	C1	C3	A
4	C2	C2	C1	C1	B
5	C2	C3	C3	C2	B
6	C2	C1	C2	C3	B
7	C2	C3	C2	C1	B
8	C2	C3	C3	C2	B
9	C3	C2	C3	C3	B
10	C3	C3	C1	C1	B
11	**C1**	**C1**	**C1**	**C1**	**A**
12	**C1**	**C2**	**C1**	**C2**	**A**
13	**C2**	**C2**	**C1**	**C3**	**A**
14	**C2**	**C2**	**C1**	**C3**	**A**

Sedangkan metode ADASYN-KNN memanfaatkan voting dari neighbor tiap data. Contoh sebagai berikut:

Tabel 9 Evaluasi majority voting data D1

NO	F1	F2	F3	F4	CLASS
Evaluasi D1	**C1**	**C1**	**C1**	**C1**	**A**
NN-1	C1	C2	C1	C2	A
NN-2	C2	C2	C1	C1	B
NN-3	C2	C2	C1	C3	A
NN-4	C2	C1	C2	C3	B
NN-5	C3	C3	C1	C1	B
VOTING	**C2**	**C2**	**C1**	**C1**	**A**

Tabel 10 Evaluasi majority voting data D2

NO	F1	F2	F3	F4	CLASS
Evaluasi D2	**C1**	**C2**	**C1**	**C2**	**A**
NN-1	C1	C1	C1	C1	A
NN-2	C2	C2	C1	C3	A
NN-3	C2	C2	C1	C1	B
NN-4	C2	C1	C2	C3	B
NN-5	C3	C3	C1	C1	B
VOTING	**C2**	**C1**	**C1**	**C1**	**A**

Tabel 11 Evaluasi majority voting data D3

NO	F1	F2	F3	F4	CLASS
Evaluasi D3	**C2**	**C2**	**C1**	**C3**	**A**
NN-1	C2	C2	C1	C1	B
NN-2	C3	C3	C1	C1	B
NN-3	C2	C1	C2	C3	B
NN-4	C3	C2	C3	C3	B
NN-5	C2	C3	C3	C2	B
VOTING	**C2**	**C2**	**C1**	**C1**	**A**

Data sintetis yang dihasilkan dari proses tersebut kemudian ditambahkan pada dataset asli sehingga membentuk dataset baru.

Tabel 12 Hasil penambahan dataset asli dengan data sintesis

No	*F1*	*F2*	*F3*	*F4*	*class*
1	C1	C1	C1	C1	A
2	C1	C2	C1	C2	A
3	C2	C2	C1	C3	A
4	C2	C2	C1	C1	B
5	C2	C3	C3	C2	B
6	C2	C1	C2	C3	B
7	C2	C3	C2	C1	B
8	C2	C3	C3	C2	B
9	C3	C2	C3	C3	B
10	C3	C3	C1	C1	B
11	C2	C2	C1	C1	A
12	C2	C1	C1	C1	A
13	C2	C2	C1	C1	A
14	C2	C2	C1	C1	A

7 Pengujian Hasil Perhitungan

Sebuah dataset dengan fitur *nominal-multi categories* diujikan menggunakan perhitungan di atas. Dataset yang diuji adalah dataset Balance-Scale yang memiliki 626 *instances* dengan 4 fitur nominal dan terdistribusi ke dalam 3 kelas, yaitu kelas Left dan Right dengan masing-masing 288 *instances* dan kelas Balance dengan 49 *instances*. Masing-masing fitur pada dataset tersebut memiliki 5 kategori. Adapun metadata dari dataset tersebut adalah sebagai berikut:

- class: L, B, R
- Left-Weight: 1, 2, 3, 4, 5
- Left-Distance: 1, 2, 3, 4, 5
- Right-Weight: 1, 2, 3, 4, 5
- Right-Distance: 1, 2, 3, 4, 5

Selanjutnya dataset Balance-Scale di-*oversampling* dengan metode ADASYN-N dan ADASYN-kNN. Penghitungan pada prosedur 2a sampai dengan 2d menunjukkan bahwa kedua metode baik ADASYN-N maupun ADASYN-kNN menghasilkan 240 data sintetis untuk menyeimbangkan kelas Balance dengan kelas lainnya. Data sintetis tersebut kemudian digabungkan dengan dataset asli dan diuji klasifikasi dengan metode Random Forest.

Implementasi dengan *classifier* tersebut dilakukan menggunakan 10-*Cross Fold Validation* dan dilakukan sebanyak 30 kali. Yang dimaksud dengan 10-*Cross Fold Validation,* yaitu membagi dataset menjadi 10 bagian, dimana satu bagian akan menjadi *testing set* dan sembilan bagian sisanya digunakan sebagai *training set*, hal ini dilakukan bergantian sebanyak sepuluh kali.

Selanjutnya, akurasi hasil klasifikasi Random Forest dibandingkan antara dataset asli dengan dataset hasil oversampling ADASYN-N dan ADASYN-KNN. Hasil perbandingan akurasi ditunjukkan pada Tabel 13.

Tabel 13 Hasil Pengujian

Deskripsi	*Nilai*
n	30
μ akurasi pada Dataset Asli	81,37%
μ akurasi dgn ADASYN-N	90,42%
μ akurasi dgn ADASYN-kNN	89,21%

Dari hasil pada Tabel 13 menunjukkan bahwa teknik *oversampling* ADASYN-N dapat meningkatkan akurasi klasifikasi sebanyak 9,05% dari dataset asli, dan ADASYN-kNN meningkatkan akurasi klasifikasi sebanyak 7,84% dari dataset asli. Hal tersebut juga menunjukkan bahwa teknik ADASYN-N lebih baik daripada ADASYN-kNN dalam menangani ketidakseimbangan pada data dengan fitur *nominal-multi categories*.

8 Kesimpulan

Dari hasil pembahasan di atas, dapat disimpulkan bahwa perhitungan kNN pada data dengan fitur *nominal-multi categories* untuk teknik *oversampling* ADASYN-kNN dan ADASYN-N menunjukkan peningkatan akurasi yang cukup signifikan dari dataset asli yang belum dilakukan proses *resampling*. Teknik ADASYN-N menunjukkan akurasi yang lebih baik dari ADASYN-kNN dalam mengatasi ketidakseimbangan distribusi kelas pada data dengan fitur *nominal-multi categories*. Berbeda dengan hasil penelitian sebelumnya yang menunjukkan bahwa akurasi pada dataset dengan teknik ADASYN-kNN lebih baik daripada teknik ADASYN-N dalam penanganan data dengan fitur *nominal-binary* (hanya terdapat dua kategori pada masing-masing fitur).

9 Referensi

[1] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adaptive Synthetic Sampling Approach for Imbalanced Learning," no. 3, pp. 1322–1328, 2008.
[2] Y. E. Kurniawati, "Multiclass Imbalance Learning dengan Adaptive Synthetic-Nominal (ADASYN-N) dan Adaptive Synthetic-KNN (Adasyn-KNN) untuk Resampling Data pada Data Hasil Tes Pap Smear," Tesis pada Departemen Teknik Elektro dan Teknologi Informasi, Fakultas Teknik, Universitas Gadjah Mada, 2017.
[3] N. Chawla and K. Bowyer, "SMOTE: Synthetic Minority Over-sampling Technique Nitesh," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.
[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.
[5] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[6] H. He and Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications, 1st ed. Wiley-IEEE Press, 2013.