LOGIC: Jurnal Penelitian Informatika
Vol. 3, Issue 1, pp. 1–5 (2025)
doi: http://doi.org/10.25124/logic.v3i1.6467

RESEARCH ARTICLE

Prediksi *Retweet* Berdasarkan Konten Dan Pengguna Dengan Metode *Classifier Selection*

Muhamad Febiansyah, Jondri* and Indwiarti

Fakultas Informatika, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia
*Corresponding author: jondri@telkomuniversity.ac.id

Abstrak

Perkembangan media sosial telah merubah cara penyebaran informasi, dengan *Twitter* memainkan peran utama. Penelitian ini bertujuan mengembangkan model prediksi *retweet* di *Twitter* menggunakan fitur *content-based* dan *user-based*, serta teknik *oversampling* untuk meningkatkan kinerja model. Hasil eksperimen menunjukkan bahwa *meta learner* tanpa *oversampling* pada fitur *content-based* memiliki *macro average F1-score* sebesar 0,52, namun dengan *recall* yang sangat rendah untuk kelas *retweet* (6%) dan *F1-score* 0,11. Sebaliknya, *meta learner* dengan *oversampling* pada fitur *content-based* memperbaiki performa dengan *presisi* 0,86, *recall* 0,77, dan *F1-score* 0,80 untuk *retweet*, dengan nilai *macro average F1-score* sebesar 0,82 yang menunjukan kenaikan dibandingkan dengan *meta learner* tanpa *oversampling*. Untuk model *user-based*, tanpa *oversampling*, *macro average F1-score* memiliki nilai 0,75 dengan keseimbangan baik antara presisi dan *recall* pada kelas *non retweet*. Setelah *oversampling*, model *user-based* mempertahankan keseimbangan yang baik dengan *presisi*, *recall*, *F1-score*, dan *macro average F1-score* masing-masing sebesar 0,88 pada kelas *retweet* dan *non retweet*. Secara keseluruhan, *oversampling* meningkatkan kinerja model, terutama pada fitur *content-based*, dengan model *user-based* menunjukkan performa yang paling konsisten dan baik.

Key words: *twitter*, pemilihan pengklasifikasi, berbasis pengguna, berbasis konten.

Pendahuluan

Perkembangan media sosial mempercepat penyebaran informasi, termasuk informasi terkait *COVID-19*. Pada Januari 2022, pengguna aktif media sosial di Indonesia mencapai 191 juta, meningkat 12,35% dari tahun sebelumnya. *Twitter* menjadi salah satu platform populer dengan lebih dari 500 juta pengguna global dan 340 juta *retweet* setiap hari [1][2]. Melalui tweet, pengguna dapat berbagi informasi berupa foto, teks, video, dan suara secara *real-time*, serta memposting ulang konten dari pengguna lain [3]. Namun, tidak semua tweet terkait *COVID-19* mendapatkan *retweet*. Membuat model prediksi untuk *retweet* sangat penting karena *retweet* berperan signifikan dalam memperluas jangkauan dan dampak dari sebuah pesan, terutama selama pandemi dimana informasi yang tepat waktu dan akurat sangat dibutuhkan. Memahami pola *retweet* dapat membantu dalam menyebarkan informasi kesehatan yang kritis lebih efektif, sehingga dapat mendukung upaya penanggulangan pandemi dan pengambilan keputusan oleh masyarakat. Dengan adanya model prediksi *retweet* yang akurat, dapat diidentifikasi faktor-faktor yang membuat suatu informasi lebih mungkin untuk tersebar luas [4].

Penelitian sebelumnya telah menggunakan berbagai metode *machine learning* seperti *Naïve Bayes*, *Fuzzy*, SVM, dan *Decision Tree* untuk prediksi *retweet*, namun hasilnya masih kurang memadai 5. Kelemahan dari metode tersebut terletak pada kemampuannya yang terbatas dalam menangani kompleksitas dan variasi data, seperti interaksi pengguna, waktu posting, dan konten *tweet*. Misalnya, penelitian yang menggunakan *Naïve Bayes* sering kali mengasumsikan bahwa fitur-fitur independen, yang tidak selalu mencerminkan kenyataan. Metode *Fuzzy* dan SVM juga menunjukkan keterbatasan dalam menangkap pola *non-linear* dalam data, sementara *Decision Tree* rentan terhadap *overfitting* ketika digunakan pada *dataset* yang lebih besar dan kompleks. Oleh karena itu, diperlukan pendekatan baru yang lebih mampu menangani kompleksitas ini untuk meningkatkan akurasi prediksi *retweet*.

Penelitian ini akan menggunakan metode *classifier selection*, yang menggabungkan beberapa algoritma *machine learning* untuk mencapai prediksi yang lebih akurat. *Base model* yang dipilih adalah *Support Vector Machine* (SVM), *Decision Tree* (DT), dan *Logistic Regression* (LR) karena ketiganya memiliki kekuatan komplementer: SVM efektif dalam menangani data *non-linear* dan tinggi dimensi, *Decision Tree* baik dalam menangkap hubungan *nonlinear* serta interpretasi model, dan *Logistic Regression* cocok untuk situasi di mana interpretabilitas model dibutuhkan serta mampu memberikan probabilitas *output*. *Meta learner* yang dipilih adalah SVM, karena kemampuannya yang kuat dalam mengklasifikasikan data kompleks setelah menerima input dari *base models*, sehingga dapat menggabungkan kekuatan dari ketiga model dasar tersebut untuk mencapai hasil prediksi yang lebih akurat

dan andal. Dengan *classifier selection* ini, diharapkan dapat ditemukan algoritma yang paling efektif dalam memanfaatkan fitur *content-based* dan *user-based*, sehingga dapat meningkatkan akurasi prediksi *retweet*, khususnya pada tweet terkait *COVID-19* [6].

Topik dan Batasannya

Penelitian ini berfokus pada prediksi *retweet* di *Twitter* dengan menggunakan fitur berbasis konten (content- based) dan berbasis pengguna (*user-based*). Model prediksi dibangun dengan metode *classifier selection*, yang mengombinasikan beberapa algoritma pembelajaran mesin. Batasan penelitian termasuk penggunaan teknik *oversampling* untuk menangani ketidakseimbangan kelas dan evaluasi model dengan metrik klasifikasi biner. Penelitian ini terbatas pada data yang dikumpulkan dari *Twitter* melalui API dan tidak mencakup faktor-faktor lain seperti sentimen atau waktu pengiriman tweet yang mungkin memengaruhi prediksi *retweet*.

Tujuan

Penelitian ini bertujuan untuk mengembangkan model prediksi *retweet* yang lebih akurat dengan memanfaatkan fitur *content-based* dan *user-based*. Penelitian ini juga bertujuan untuk mengeksplorasi efektivitas teknik *oversampling* dalam meningkatkan kinerja model, terutama dalam menghadapi ketidakseimbangan kelas. Dengan menggunakan pendekatan *meta learner* dan *classifier selection*, penelitian ini bertujuan untuk menemukan kombinasi algoritma yang optimal untuk memprediksi *retweet*, serta mengevaluasi dampak teknik *oversampling* terhadap kinerja prediksi.

Organisasi Tulisan

Penulisan dimulai dengan melakukan tinjauan literatur yang mencakup berbagai topik. Selanjutnya, metodologi yang digunakan dalam penelitian ini akan dijelaskan. Pada tahap berikutnya, hasil penelitian akan dievaluasi dan dibahas. Terakhir, kesimpulan dan saran akan disampaikan.

Tinjauan Pustaka

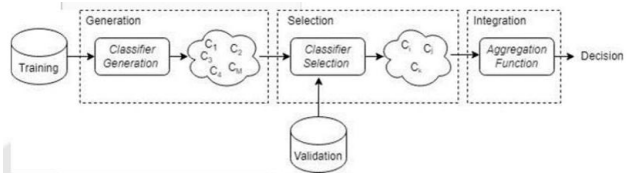
Twitter

Twitter merupakan salah satu media sosial yang memungkinkan berbagai aktivitas seperti memposting foto, video, suara, dan teks, serta memposting ulang informasi dari pengguna lain [7]. Informasi di *Twitter* dapat menyebar dengan cepat karena adanya fitur *retweet* yang sering digunakan oleh pengguna. Secara struktur, fitur *retweet* mirip dengan penggunaan *email*, di mana pengguna dapat mengirim ulang *email* yang diterima dari orang lain. Oleh karena itu, fitur *retweet* memungkinkan penyebaran informasi yang lebih luas dan dapat dipahami oleh pengguna lain [8].

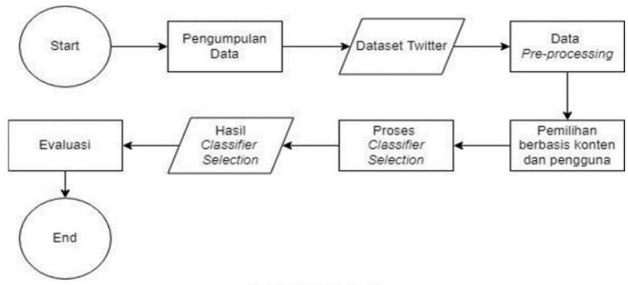
Selection Feature

Untuk melakukan penelitian, perlu dilakukan pemilihan fitur yang bertujuan untuk mendapatkan hasil prediksi yang diinginkan. Pemilihan fitur dapat dilakukan setelah pengumpulan data dan *preprocessing*. Fitur yang akan digunakan dalam penelitian ini adalah *content-based* dan *user-based*.

1.Atribut Data
a.*author*: penulis *tweet*.
b.deskripsi: isi dari konten.
c.lokasi: lokasi yang disebutkan pada isi *tweet*.
d.favorit: *tweet* yang dimasukkan ke dalam favorit.
e.jumlah_*retweet*: apakah *tweet* telah di*retweet*.
f.sentiment: apakah *tweet* memiliki sentiment negatif atau positif.



Gambar 1. Arsitektur *Classifier Selection*.



Gambar 2. *Flowchart*

2.*Content Based*
Content based merupakan fitur untuk memfilter konten, dimana *system* akan memberikan rekomendasi kepada pengguna berdasarkan aktivitas pengguna tersebut[9]
3.*User Based*
User based merupakan fitur untuk memproses pemberian rating oleh pengguna lain terhadap suatu informasi dengan menggunakan *cosine similarity* antar pengguna[10]. Fitur ini didasarkan pada interaksi antar satu pengguna dengan pengguna lainnya sehingga menjadi penting untuk diperhatikan dalam penelitian.

Classifier Selection

Classifier selection adalah cara untuk memilih model terbaik dalam menyelesaikan masalah. Dalam proses ini, mesin pembelajaran akan mencoba memprediksi data uji seakurat mungkin untuk hasil terbaik. Prosesnya terdiri dari tiga tahap: generasi, seleksi, dan integrasi. Pada tahap generasi, beberapa model dasar dibangun dengan berbagai strategi seperti ditunjukan pada gambar 1. Tahap seleksi melibatkan pemilihan model terbaik berdasarkan kriteria yang ditentukan menggunakan data validasi. Tahap terakhir adalah integrasi, dimana *output* dari model terpilih digabungkan sesuai dengan aturan yang telah ditetapkan [11].

Metodologi Penelitian

Gambar 2 merupakan rancangan *system* dari prediksi *retweet* berbasis konten dan pengguna dengan metode *classifier selection*. Tahap ini meliputi pengumpulan data, *preprocessing*, pemilihan berbasis konten dan pengguna, *classifier selection*, dan berakhir pada evaluasi.

Dataset

Dataset ini dikumpulkan dari *Twitter* menggunakan *Twitter* API, yang tersedia untuk pengguna terdaftar sebagai developer. *Dataset* ini terdiri dari 1.275 baris dan 34 kolom, mencakup berbagai informasi seperti akun pengguna, konten *tweet*, dan sentimen. Untuk analisis ini, *dataset* dibagi menjadi dua subset: 60% digunakan sebagai *data train* dan 40% sebagai *data test*. Pembagian ini memungkinkan pengujian model dengan data yang tidak terlihat sebelumnya untuk mengevaluasi kinerjanya.

Preprocessing

Preprocessing merupakan tahap yang dilakukan setelah pengumpulan data, pada data yang akan di gunakan, perlu dilakukan penyeleksian kata yang ada pada *tweets* sehingga menghasilkan kata-kata yang lebih terstruktur. *Preprocessing* dilakukan dengan berbagai tahap, yaitu:

1. Mengatasi *missing value* yang ada pada *dataset* yang digunakan.
2. Mengecek kembali apakah data yang ada mempunyai nilai *duplicate*.
3. Menghapus data yang memiliki nilai *duplicate*
4. Mengecek apakah ada *outliner* pada *dataset*.
5. Menghapus *outliner* yang ada pada *dataset*.
6. Mengecek *imbalance class*, dimana 0 merupakan kelas yang tidak mendapatkan *retweet*, sedangkan 1 merupakan kelas yang mendapatkan *retweet*.

Classification

1. *Base Learner*
   Pada tingkat satu *classifier selection* yang nantinya akan disusun memiliki tiga metode *classifikasi* untuk bagian *base-learner* yaitu:
   a. SVM
      SVM (*Support Vector Machine*) adalah proses tipe supervisi dalam pembelajaran mesin yang menganalisis dan mengidentifikasi pola dalam data input untuk melakukan klasifikasi atau analisis regresi. SVM digunakan dalam berbagai aplikasi, seperti pengenalan angka, pengenalan tulisan tangan, deteksi wajah, klasifikasi kanker, peramalan deret waktu, dan lainlain[12].
   b. *Decision Tree*
      Decision tree adalah model prediktif dalam pembelajaran mesin yang digunakan untuk klasifikasi atau regresi. Model ini membagi data ke dalam subset berdasarkan fitur tertentu hingga mencapai hasil akhir. *Decision tree* diilustrasikan sebagai struktur pohon, di mana setiap simpul internal adalah fitur, setiap cabang adalah aturan keputusan, dan setiap simpul daun adalah hasil atau label[13].
   c. *Logistic Regression*
      Logistic Regression adalah salah satu metode klasifikasi yang umum digunakan dalam analisis data. Dalam konteks *Machine Learning*, *Logistic Regression* adalah salah satu algoritma yang sering digunakan untuk masalah klasifikasi biner, dimana tujuan utamanya adalah untuk memprediksi probabilitas bahwa suatu *instance* tertentu termasuk dalam kelas tertentu[14].

2. *Meta Learner*
   Meta learner adalah pendekatan dalam *Machine Learning* di mana algoritma mempelajari dari berbagai tugas atau *dataset* untuk mempercepat pembelajaran pada tugas atau *dataset* baru. Dalam penelitian ini, kami akan menggunakan meta-learning untuk mengoptimalkan proses klasifikasi dengan menggunakan *Support Vector Machine* (SVM) sebagai classifier pada tingkat kedua. Meta-learner kami akan dilatih dengan hasil prediksi dari beberapa metode dasar, sehingga dapat menghasilkan akurasi yang baik[15].

Eksperimen

Dalam penelitian ini, eksperimen dilakukan dengan menerapkan teknik *oversampling* pada *dataset* yang mengalami ketidakseimbangan kelas. *Oversampling* adalah strategi yang sering digunakan dalam pengolahan data untuk mengatasi masalah ketidakseimbangan kelas dengan meningkatkan jumlah sampel dari kelas minoritas[16]. Pada kode yang digunakan, teknik SMOTE (*Synthetic Minority Over-sampling Technique*) diterapkan untuk menghasilkan sampel sintetis dari kelas minoritas, sehingga menciptakan keseimbangan yang lebih baik antara kelas-kelas dalam *dataset*. Setelah proses *oversampling*, *dataset* dibagi menjadi set pelatihan dan set pengujian dengan proporsi 60% untuk pelatihan dan 40% untuk pengujian. Pembagian ini memastikan bahwa model yang dilatih dapat belajar dari data yang lebih seimbang antara kelas minoritas dan kelas mayoritas. Distribusi label setelah *oversampling* ditampilkan untuk memverifikasi bahwa jumlah *instance* dari setiap kelas telah diperbaiki sesuai dengan tujuan *oversampling*.

Table 1. *Confusion Matrix*

	Actual Positif	**Actual Negatif**
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN	False Negative (FN

Evaluasi

Evaluasi model digunakan untuk mengevaluasi kinerja *system* yang telah dirancang, yang nantinya akan menggunakan *binary classification metrics*. Di dalam *binary classification metrics* terdapat berbagai macam perhitungan performasi, salah satunya adalah *confusion metrix*. *Confusion metrix* akan menghasilkan perhitungan berupa akurasi, presisi, *recall*, dan *F1Measure*.

Hasil dan Pembahasan

Evaluasi model digunakan untuk mengevaluasi kinerja *system* yang telah dirancang, yang nantinya akan menggunakan *binary classification metrics*. Di dalam *binary classification metrics* terdapat berbagai macam perhitungan performasi, salah satunya adalah *confusion metrix*. *Confusion metrix* akan menghasilkan perhitungan berupa akurasi, presisi, *recall*, dan *F1Measure*. Berikut adalah table dari *confusion matrix* 1 :

1. Akurasi
   Akurasi merupakan hasil yang menunjukkan seberapa akurat sebuah *system* yang telah dibuat dalam melakukan klasifikasi dengan benar. Berikut rumus dari akurasi:

$$Accuracy = \frac{\text{TP + TN}}{\text{Total Number Of Data}} \tag{1}$$

2. Presisi
   Presisi merupakan hasil yang menunjukkan perbandingan jumlah sampel yang diprediksi berada dikelas yang benar dan jumlah sampel yang diprediksi oleh sistem klasifikasi. Berikut rumus dari presisi:

$$Precision = \frac{\text{TP}}{\text{TP + FP}} \tag{2}$$

3. *Recall*
   Recall merupakan hasil yang menunjukkan rasio jumlah sampel yang diprediksi dengan benar dengan jumlah yang seharusnya diprediksi. Berikut rumus dari *recall* :

$$Recall = \frac{\text{TP}}{\text{TP + FN}} \tag{3}$$

4. *F1-Measure*
   F1-Measure merupakan hasil yang menunjukkan pengukuran untuk analisis kinerja klasifikasi. Berikut rumus dari *F1-Measure*:

$$F1 - Measure = \frac{\text{2x Precision xRecall}}{\text{Precision + Recall}} \tag{4}$$

Table 2. *Content Based Meta Learner*

	Precision	***Recall***	***F1-Score***	**Support**
0	0.88	1.00	0.93	3448
1	0.71	0.06	0.11	503
Macro Avg F1-Score			0.52	3951

Table 3. *User Based Meta Learner*

	Precision	***Recall***	**F1-Score**	**Support**
0	0.94	0.99	0.96	1815
1	0.84	0.40	0.54	1286
Macro Avg F1-Score			0.97	738

Table 4. *Content Based Meta Learner dengan Oversampling* Data

	Precision	***Recall***	***F1-Score***	**Support**
0	0.79	0.87	0.83	3454
1	0.86	0.77	0.81	3460
Macro Avg F1-Score			0.82	6914

Hasil Pengujian

Dalam penelitian ini, dibuat beberapa fungsi untuk melatih *dataset*, seperti SVM, *Decision Tree*, dan *Logistic Regresion* menggunakan *base learner*. *Base learner* merupakan fungsi *cross validation* untuk menghasilkan probabilitas dari masing masing model. Kemudian hasil prediksi akan digunakan untuk melatih fungsi *meta learner*, kemudian akan menghasilkan model *meta learner* yang telah dilatih. Lalu, terdapat fungsi classifier yang akan melatih *base learner* dan *meta learner* yang mana nanti akan dihitung hasil pemodelan dengan *confusion matrix*. Di percobaan awal, dilakukan beberapa eksperimen untuk mencari model terbaik diantaranya *meta learner* tanpa *oversampling* data, dan *meta learner* dengan *oversampling* data.

1. *Meta Learner Result*

a. *Content Based*

Hasil evaluasi dari *meta learner* terhadap metode berbasis konten terbagi menjadi dua kategori: 0 untuk *non-retweet* dan 1 untuk *retweet*. Evaluasi terhadap model berbasis konten menunjukkan kinerja yang cukup baik dengan *macro average F1-score* 0.52. Meskipun model ini memiliki nilai *precision* yang tinggi untuk kelas *non-retweet* (0.88), model ini sangat kurang efektif dalam mendeteksi *retweet* (kelas 1), dengan *recall* yang sangat rendah yaitu hanya 0.06. Artinya, model hanya mampu mendeteksi 6% dari seluruh *instance retweet* yang ada, sementara 94% *retweet* tidak terdeteksi dengan benar. Selain itu, nilai *F1-score* yang tinggi (0.93) hanya tercapai pada kelas *non-retweet*, sedangkan *F1-score* untuk kelas *retweet* hanya sebesar 0.11. Nilai *macro average* menunjukkan bahwa meskipun model ini efektif dalam mendeteksi kelas *nonretweet*, performa pada kelas *retweet* sangat buruk, menghasilkan ketidakseimbangan yang signifikan antara presisi dan *recall*. *Macro average* memberikan gambaran yang lebih komprehensif tentang kinerja model pada kedua kelas secara keseluruhan, dan dalam hal ini, menunjukkan bahwa model mengalami kesulitan yang signifikan dalam mendeteksi *retweet*. Oleh karena itu, meskipun akurasi keseluruhan tampak baik, *macro average* menyoroti ketidakseimbangan yang signifikan dalam kinerja model, terutama dalam mendeteksi *retweet*. Berikut adalah hasil penelitian *meta learner* terhadap *content based* 2 :

b. *User Based*

Hasil evaluasi dari *meta learner* terhadap metode berbasis *user-based* terbagi menjadi dua kategori: 0 untuk *non-retweet* dan 1 untuk *retweet*. Evaluasi terhadap model *user-based* menunjukkan kinerja yang sangat baik dengan *macro average F1-Score* sebesar 0,75. Model ini berhasil mencapai nilai *precision* 0.94, *f1-score* dengan nilai 0.96, dan *recall* 0.99 untuk kelas *non-retweet*, yang menunjukkan bahwa model sangat efektif dalam mendeteksi hampir semua *instance nonretweet* dengan tingkat kesalahan prediksi yang sangat rendah, yaitu hanya sekitar 1%. Namun, untuk kelas *retweet*, model menunjukkan performa yang kurang optimal, dengan *precision* 0.84, *recall* 0.40, dan *F1-score* 0.54. Meskipun *precision* untuk kelas *retweet* relatif baik, *recall* yang rendah menunjukkan bahwa model hanya mampu mendeteksi 40% dari seluruh *instance retweet* yang ada, sementara 60% *retweet* tidak terdeteksi dengan benar. *Macro average* memberikan gambaran menyeluruh tentang kinerja model di kedua kelas, mencerminkan keseimbangan performa yang lebih baik meskipun terdapat perbedaan signifikan antara deteksi kelas *non-retweet* dan *retweet*. Secara keseluruhan, model *user-based* sangat andal dalam mengidentifikasi *non-retweet*, tetapi masih membutuhkan perbaikan dalam mendeteksi *retweet*. Berikut adalah hasil penelitian *meta learner* terhadap *user-based* 3:

2. *Meta Learner With Oversampling Data Result*

Setelah dilakukan *oversampling*, jumlah data untuk *content based* menjadi 6914 data dan *userbased* menjadi 3617 data.

a. *Content Based*

Hasil evaluasi dari *meta learner* dengan *oversampling* data terhadap metode berbasis konten menunjukkan perbaikan kinerja yang signifikan. Untuk kelas *non-retweet*, model ini menghasilkan *precision* sebesar 0.79, yang menunjukkan bahwa 79% dari prediksi yang diklasifikasikan sebagai *non-retweet* adalah benar. Nilai *recall* sebesar 0.87 menunjukkan bahwa model mampu mendeteksi 87% dari *instance non-retweet* yang sebenarnya. Meskipun *F1-score* sebesar 0.83 pada kelas *non-retweet* menunjukkan kinerja yang baik, masih terdapat ketidakseimbangan antara presisi dan *recall*, seperti yang tercermin dari nilai *precision* dan *recall*. Untuk kelas *retweet*, model menunjukkan performa yang lebih baik dengan *precision* sebesar 0.86 dan *recall* sebesar 0.77. Ini mengindikasikan bahwa model lebih efektif dalam mendeteksi *retweet* dibandingkan dengan *non-retweet*, dengan *F1-score* sebesar 0.81 yang menunjukkan keseimbangan yang lebih baik antara presisi dan *recall*.

Penerapan *oversampling* data telah membantu meningkatkan kinerja model dalam mendeteksi *retweet* dibandingkan dengan pendekatan sebelumnya. *Macro average F1-score* sebesar 0.82 mencerminkan keseimbangan yang baik antara performa model pada kedua kelas, menandakan bahwa *oversampling* data telah memberikan kontribusi positif terhadap keseimbangan antara deteksi *non-retweet* dan *retweet*. Berikut adalah hasil penelitian *meta learner with oversampling* data terhadap *content-based* 4:

b. *User Based*

Hasil penelitian yang menggunakan *meta learner* dengan *oversampling* data pada model *user-based* menunjukkan kinerja yang sangat konsisten dan seimbang dalam mendeteksi baik *retweet* (kelas 1) maupun *non-retweet* (kelas 0). Model ini mencapai nilai *precision*, *recall*, dan *F1-score* yang seragam, yaitu sebesar 0.88 untuk kedua kelas. Untuk kelas *nonretweet* (kelas 0), model ini memiliki *precision* dan *recall* masing-masing sebesar 0.88, menunjukkan bahwa 88% dari prediksi *non-retweet* adalah benar dan model berhasil mendeteksi 88% dari seluruh *instance non-retweet* yang ada. *F1-score* yang juga sebesar 0.88 menunjukkan keseimbangan yang baik antara *precision* dan *recall*, yang menunjukkan kinerja model yang konsisten dalam mendeteksi *non-retweet*. Untuk

Table 5. *User Based Meta Learner dengan Oversampling Data*

	Precision	***Recall***	***F1-Score***	**Support**
0	0.88	0.88	0.88	1804
1	0.88	0.88	0.88	1813
Macro Avg F1-Score			0. 88	3617

kelas *retweet* (kelas 1), model ini juga menunjukkan nilai *precision* dan *recall* sebesar 0.88, yang berarti model dapat mendeteksi 88% dari *instance retweet* dengan tingkat akurasi yang sama dengan kelas *non-retweet*. *F1-score* sebesar 0.88 pada kelas *retweet* menegaskan bahwa model ini mencapai keseimbangan yang efektif antara *precision* dan *recall*. Dengan *macro average F1-score* sebesar 0.88, hasil ini mencerminkan performa yang sangat baik di kedua kelas, tanpa kecenderungan bias terhadap salah satu kelas. Berikut adalah hasil penelitian *meta learner* dengan *oversampling* data terhadap *user-based* 5:

Kesimpulan

Berdasarkan hasil pada Bab IV, penelitian ini menunjukkan bahwa penggunaan *meta learner* dan teknik *oversampling* memiliki dampak signifikan terhadap kinerja model prediksi *retweet*. Pada tahap awal, *meta learner* tanpa *oversampling* menunjukkan kinerja baik pada model *content-based* dengan *macro average F1-score* sebesar 0.52. Namun, model ini menghadapi kesulitan dalam mendeteksi *retweet*, dengan *recall* yang sangat rendah dan *F1-score* yang tidak seimbang antara kelas *retweet* dan *non-retweet*. Sebaliknya, model *user-based* tanpa *oversampling* menunjukkan hasil lebih baik, dengan *macro average F1-score* sebesar 0.75 dan keseimbangan yang lebih baik antara presisi dan *recall*, terutama dalam mendeteksi kelas *retweet*. Setelah penerapan teknik *oversampling*, terjadi peningkatan signifikan pada beberapa metrik, terutama dalam mendeteksi *retweet*. Pada model *content-based* dengan *oversampling*, presisi dan *recall* untuk kelas *retweet* meningkat menjadi 0.86 dan 0.77, dengan *F1-score* 0.80, menunjukkan peningkatan jelas dalam kemampuan model untuk mendeteksi *retweet*. Meskipun ada perbaikan, model ini masih menunjukkan ketidakseimbangan antara presisi dan *recall* pada kelas *non-retweet*.

Model *user-based* dengan *oversampling* menunjukkan hasil sangat baik, dengan presisi, *recall*, *F1-score*, dan *macro average F1-score* seragam sebesar 0.88 untuk kedua kelas, mencerminkan kinerja yang lebih seimbang dan andal. Secara keseluruhan, penerapan teknik *oversampling* pada *meta learner* terbukti meningkatkan kinerja model, terutama dalam mendeteksi *retweet*. Namun, tantangan dalam menjaga keseimbangan antara presisi dan *recall* masih perlu diatasi, khususnya pada model *content-based*.

Daftar Pustaka

1. Data Indonesia. DataIndonesia.id; 2022. Accessed: 25 February 2022. Available from: `https://dataindonesia.id/`.
2. Luo Z, Osborne M, Tang J, Wang T. Who Will Retweet Me? Finding Retweeters in Twitter. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2013. p. 869-72.
3. Firdaus SN, Ding C, Sadeghian A. Retweet Prediction Considering User's Difference as an Author and Retweeter. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE; 2016. p. 852-9.
4. Hoang TBN, Mothe J. Predicting Information Diffusion on Twitter–Analysis of Predictive Features. Journal of Computational Science. 2018;28:257-64.
5. Dong X, Yu Z, Cao W, Shi Y, Ma Q. A Survey on Ensemble Learning. Frontiers of Computer Science. 2020;14(2):241-58.
6. Khan I, Zhang X, Rehman M, Ali R. A Literature Survey and Empirical Study of Metalearning for Classifier Selection. IEEE Access. 2020;8:10262-81.
7. Firdaus SN, Ding C, Sadeghian A. Retweet: A Popular Information Diffusion Mechanism–A Survey Paper. Online Social Networks and Media. 2018;6:26-40.
8. Boyd D, Golder S, Lotan G. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. In: 2010 43rd Hawaii International Conference on System Sciences. IEEE; 2010. p. 1-10.
9. Utami FA. Apa Itu Content-Based Filtering; 2022. Accessed 18 May 2022. `https://wartaekonomi.co.id/read412507/apa-itucontent-based-filtering`.
10. Nugraha D, Purboyo TW, Nugrahaeni RA. Sistem Rekomendasi Film Menggunakan Metode User Based Collaborative Filtering. eProceedings of Engineering. 2021;8(5).
11. Suyanto, Arifianto A, Rismala R, Sunyoto A. Evolutionary Machine Learning (Pembelajaran Mesin Otonom Berbasis Komputasi Evolusioner). Informatika; 2020.
12. Behera MP, Sarangi A, Mishra D, Sarangi SK. A Hybrid Machine Learning Algorithm for Heart and Liver Disease Prediction Using Modified Particle Swarm Optimization with Support Vector Machine. Procedia Computer Science. 2023;218:818-27.
13. Abdulazeez AM, Brifcani A, Issa AS. Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends. 2021;2(1):21-46.
14. Smith J, Johnson M, Williams R. Application of Logistic Regression in Health Data Classification: A Machine Learning Approach. Journal of Health Informatics. 2018;10(2):87-95.
15. Vanschoren J, et al. Meta-Learning: A Survey. arXiv preprint. 2018;arXiv:1810.03548. Available from: `https://arxiv.org/abs/1810.03548`.
16. Islam MAK, Islam MM, Shahriar MS, Alam MR. A Comprehensive Review on Class Imbalance Problem: Dataset Characteristics, Oversampling Methods, and Their Effects. Journal of Machine Learning Research. 2021;22(3):567-89.