Prediksi *Retweet* Menggunakan Fitur Berbasis Pengguna dan Fitur Berbasis Konten dengan Metode Klasifikasi ANN

Hamidan Amarullah Purwaatmaja Ash-Shidiq EFSA[1], Jondri[2], Kemas Muslim Lhaksmana[3]

[1,2,3] Universitas Telkom, Bandung
[1]hamidanamarullah@students.telkomuniversity.ac.id, [2]jondri@telkomuniversity.ac.id, [3]kemasmuslim@telkomuniversity.ac.id

Abstrak

***Tweet* merupakan sebuah pesan berisikan informasi yang dibagikan oleh pengguna Twitter. *Tweet* dapat dibagikan kepada pengguna lain dengan cara *retweet,* sehingga fitur ini berperan sangatpenting dalam penyebaran informasi. Penelitian ini membahas prediksi *retweet* menggunakan fitur berbasis pengguna dan fitur berbasis konten, serta ANN sebagai *classifier*. Masalah penelitian yang dikaji di sini adalah cara menanggulangi duplikasi data serta ketidakseimbangan kelas dengan menggunakan *undersampling* dan *oversampling*. Hasil evaluasi menunjukan bahwa proses klasifikasi mencapai nilai F1 skor 86% pada model dengan melakukan undersampling dan menghapus duplikasi pada data.**

Kata kunci: prediksi, *retweet, undersampling, oversampling*, ANN.

Abstract

Tweet is a message containing information shared by Twitter users. Tweets can be shared with other users by retweets, so this feature plays a very important role in disseminating information. This studydiscusses retweet prediction using user-based feature and content-based feature, as well as ANN as a classifier. The research problem studied here is how to overcome data duplication and class balance by using undersampling and oversampling. The evaluation results show that the classification process reachesan F1 score of 86% on the model by undersampling and removing duplication in the data.

Keywords: prediction, retweet, undersampling, oversampling, ANN.

1. Pendahuluan

Twitter adalah salah satu media sosial yang popular saat ini yang dapat digunakan untuk membagikan informasi dalam sebuah postingan yang biasa disebut sebuah *tweet*. Ada beberapa aktivitas yang dapat dilakukan ketika menggunakan Twitter. Pengguna Twitter dapat mengikuti seseorang agar mendapat informasi dari orang yang telah diikutinya dan pengguna juga dapat membuat *tweet* berupa teks, video atau gambar untuk dibagikan kepada pengikutnya [1].

Pada aplikasi Twitter terdapat fitur *retweet,* dimana pengguna dapat membagikan *tweet* pengguna lain yang disukainya untuk dibagikan kepada pengikutnya agar pengikutnya mengetahui informasi yang didapat. Praktik *retweet* ini menarik untuk dibahas, seperti fitur atau faktor apa saja yang mempengaruhi terjadinya *retweet* serta bagaimana membangun sebuah model yang dapat mengklasifikasikan kelas *retweet* dengan pengenalan pola yang ada pada data *tweet* [2,3]. Terdapat kesenjangan dalam penyebaran informasi di Twitter dimana hanya beberapa *tweet* saja yang mendapat *retweet* dibandingkan dengan *tweet* lainnya.

Puspita dkk. membandingkan model jaringan syaraf tiruan dengan *naïve bayes* untuk memprediksi kelahiran prematur. Peneliti membandingkan akurasi untuk model prediksi. Hasil penelitian menunjukan bahwa jaringan syaraf tiruan menghasilkan nilai akurasi sebesar 90.67%, sedangkan untuk *naïve bayes* menghasilkan akurasi sebesar 84.58%. Sehinga dapat disimpulkan bahwa algoritma jaringan syaraf tiruan memiliki akurasi yang unggul sebesar 6.14% dalam memprediksi kelahiran prematur [4].

Pada penelitian ini, penulis membangun model prediksi *retweet* dengan data yang digunakan sebanyak 11229 data yang diambil menggunakan Twitter API. Lalu, fitur yang digunakan adalah fitur berbasis pengguna dan fitur berbasis konten. *Tweet* yang digunakan adalah *tweet* berbahasa Indonesia. Tujuan dari tugas akhir ini adalah membangun model yang dapat memprediksi terjadinya *retweet* dari suatu *tweet*.

Metode yang diusulkan pada prediksi *retweet* ini adalah metode klasifikasi menggunakan *Artificial Neural Network* dimana ANN ini memiliki kinerja prediksi yang baik, dapat mengatasi hubungan yang kompleks dengan baik dan toleransi tinggi terhadap *noisy* data [5] serta kemampuan *adaptive learning* dan *self-organization* untuk merepresentasikan informasi selama waktu pembelajaran terhadap data yang diberikan [2].

2. Studi Terkait

Daga dkk. melakukan penelitian untuk memprediksi *like* dan *retweet* menggunakan algoritma klasifikasi seperti SVM, *Naïve Bayes*, *Logistic Regression*, *Random Forest* dan *Neural Network* dengan menggunakan 2 pendekatan pemrosesan teks yaitu TFIDF dan Doc2Vec pada data Twitter sebanyak 2 juta *tweet*. Fitur yang digunakan fitur konten (waktu tweet dibuat, jumlah *like*, jumlah *retweet* dan teks). Hasil yang diperoleh menunjukan bahwa semua model prediksi *retweet* dan prediksi *like* dengan pendekatan TFIDF memiliki nilai akurasi 10-15% lebih baik [1].

Suh dkk. meneliti bagaimana dan mengapa informasi atau *tweet* tertentu menyebar lebih luas daripada *tweet* lainnya menggunakan *Principal Components Analysis*, *Primary Components Analysis* dan *Generalized Linier Model* pada data Twitter sebanyak 74 juta *tweet*. Berdasarkan hasil penelitiannya, Suh dkk. beranggapan bahwa fitur konten yang dipakai (URL dan *hashtags*) memiliki korelasi yang kuat dengan *retweetability* serta fitur kontekstual (jumlah pengikut, usia akun dan pengikut) mempengaruhi *retweetability*. Sedangkan jumlah *tweet* tidak memprediksi *retweetability* pengguna [3].

Hoang dkk. melakukan penelitian untuk memprediksi apakah sebuah postingan *tweet* akan di-*retweet* atau tidak serta berapa banyak informasi atau *tweet* tersebut tersebar dengan menggunakan algoritma pembelajaran mesin seperti *Naïve Bayes*, *Support Vector Machine* dan *Random Forest* dengan metode validasinya yaitu 10-*Fold Cross Validation* pada data Twitter sebanyak 16 juta *tweet*. Hoang dkk. menambahkan fitur baru pada penelitiannya yang dikembangkan dari penelitian penelitian sebelumnya serta membaginya menjadi 3 kelompok yaitu fitur berbasis pengguna, berbasis waktu dan berbasis konten [6].

Selanjutnya, Zhang dkk. mengusulkan *attention-based deep neural network* untuk memasukkan informasi sosial dan kontekstual pada penelitiannya dengan menggunakan *embeddings* untuk merepresentasikan pengguna, minat pengguna, penulis *tweet* dan *tweet*. Hasil penelitian menunjukan bahwa metode yang diusulkan memiliki F1 *score* sebesar 72.1% lebih tinggi dibandingkan dengan metode lainnya yaitu 58.2% dan 69.3% yang masing masing adalah CNN *with user information* dan CNN *with user and author information* [7].

Fitur *Retweet*

Fitur *retweet* ini adalah fitur atau faktor yang mempengaruhi terjadinya *retweet*. Pada Penelitian [3], peneliti memilih fitur-fitur yang tidak berkaitan dengan bahasa agar model yang dibangun dapat digunakan oleh seluruh bahasa. Akan tetapi pada penelitian [6], peneliti menggunakan fitur yang ada pada penelitian [3] dengan menambahkan juga fitur baru yang digunakan pada model mereka. Pada penelitian [6], peneliti mengelompokan berbagai fitur *retweet* ke dalam beberapa kelompok.

Fitur Berbasis Pengguna

- Total_of_tweet, adalah total *tweet* yang sebelumnya di-*posting* pengguna di *timeline* yang memiliki tipe data numerik.
- No_of_followers, adalah jumlah orang yang mengikuti pengguna yang memiliki tipe data numerik.
- No_of_followees, adalah jumlah orang yang diikuti pengguna yang memiliki tipe data numerik.
- Age_of_account, adalah jumlah hari sejak akun dibuat yang memiliki tipe data numerik.
- No_of_favourite, adalah jumlah *tweet* yang disukai pengguna di *timeline* yang memiliki tipe data numerik.
- No_of_groups, adalah jumlah grup tempat pengguna berada yang memiliki tipe data numerik.
- Aver_favou_per_day, adalah rata-rata favorit perhari yang memiliki tipe data numerik.
- Aver_tweets_per_day, adalah rata-rata *tweet* perhari yang memiliki tipe data numerik.
- User_name_len, adalah Panjang nama pengguna yang memiliki tipe data numerik.

Fitur Berbasis Waktu

- Is_post_at hol, adalah *tweet* di-*posting* pada hari libur yang memiliki tipe data Boolean.
- Is_posted_at_noon, adalah *tweet* di-*posting* dari jam 11:00 sampai 13:00 yang memiliki tipe data Boolean.
- Is_posted_at_eve, adalah *tweet* di-*posting* dari jam 18:00 sampai 21:00 yang memiliki tipe data Boolean.
- Is_posted_at_wee, adalah *tweet* di-*posting* pada akhir pekan yang memiliki tipe data Boolean.

Fitur Berbasis Konten

- Contain_location, adalah *tweet* berisi nama lokasi yang memiliki tipe data Boolean.
- Contain_org, adalah *tweet* berisi nama organisasi yang memiliki tipe data Boolean.
- Contain_tvshow, adalah *tweet* berisi nama acara TV yang memiliki tipe data Boolean.
- Sentiment_level, adalah *tweet* diklasifikasikan ke dalam level sentimen yang memiliki tipe data { Positif, Negatif, Objektif }.

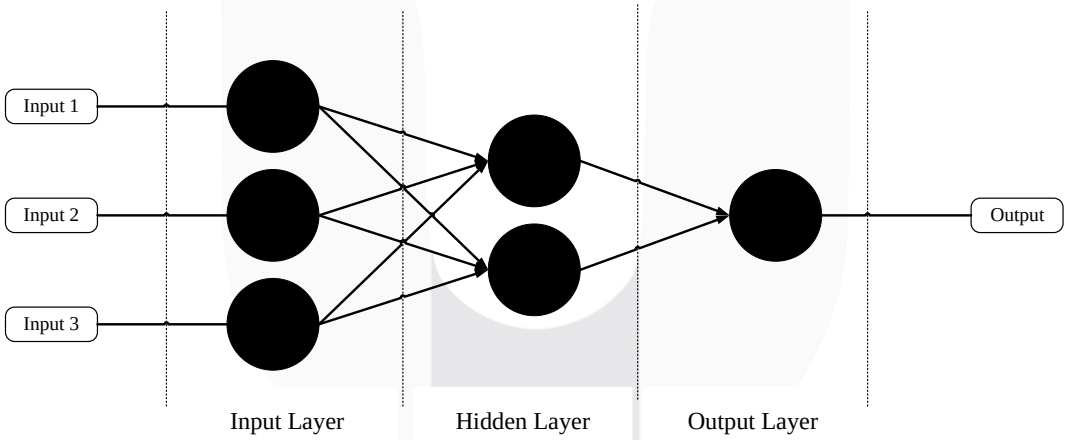
- Contain_video, adalah *tweet* berisi video yang memiliki tipe data Boolean.
- Contain_picture, adalah *tweet* berisi gambar yang memiliki tipe data Boolean.
- Contain_upper, adalah seluruh *tweet* berisi huruf besar (Huruf Kapital) yang memiliki tipe data Boolean.
- Contain_number, adalah tweet berisi nomor yang memiliki tipe data Boolean.
- Contain_excl, adalah *tweet* berisi tanda seru yang memiliki tipe data Boolean.
- Contain_rt_term, adalah *tweet* berisi atau mengandung kata "RT" yang memiliki tipe data Boolean.
- Contain_user_mentioned, adalah *tweet* menyebutkan nama pengguna lain yang memiliki tipe data Boolean.
- Contain_rt_suggest, adalah *tweet* tersebut berisi salah satu istilah saran *retweet* yang memiliki tipe data Boolean.
- Contain_url, adalah *tweet* yang mengandung *link* URL yang memiliki tipe data Boolean.
- Contain_hashtag, adalah *tweet* mengandung tagar yang memiliki tipe data Boolean.
- Opt_length, adalah panjang konten diantara 70 sampai 100 karakter yang memiliki tipe data Boolean.
- Len_of_text, adalah panjang dari konten yang memiliki tipe data numerik.

K-*Fold Cross Validation*

Merupakan prosedur pengambilan sampel data untuk mengevaluasi model *machine learning*. Lalu terdapat parameter 'k' yang menentukan berapa banyak *dataset* harus dibagi menjadi 'k' bagian. Ini adalah metode yang popular karena mudah dipahami dan biasanya menghasilkan kemampuan model dengan bias yang rendah dibandingkan dengan metode lain seperti *train test split* sederhana [8].

Artificial Neural Network

Adalah algoritma atau metode matematika yang mencoba mensimulasikan struktur dan fungsi jaringan syaraf biologis [9]. Seperti pada Gambar 1 dimana ANN memiliki 3 *layer* yaitu pada lapisan luar ada *input layer*, lalu *hidden layer* dan terakhir adalah *output layer*. ANN juga merupakan algoritma *machine learning* yang popular dan membantu untuk *classification*, *clustering*, pengenalan pola dan prediksi banyak disiplin ilmu [10][11].



Gambar 1 Struktur *Artificial Neural Network*

Backpropagation

Backpropagation (BP) merupakan *supervised machine learning algorithm* pada ANN yang digunakan untuk menentukan bobot pada setiap neuron yang dapat menghasilkan nilai *error* seminimal mungkin pada *dataset*. BP mencakup *training* & *forecast*, yang dirinci pada langkah-langkah berikut [12]:

- *Forward propagation of input signal*. Merupakan vektor masukan yang menyebar ke *output layer* setelah melakukan komputasi melalui *hidden layer*.
- *Backward propagation of error signal*. Merupakan *error propagation* secara mundur melalui jaringan neural asli, jika nilai *error* tidak memenuhi toleransi yang diberikan.
- *Weight and threshold updates*. Yaitu *update* nilai bobot dan *threshold*. Bobot dan *threshold* disesuaikan dengan *error propagation* sampai nilai *error* memenuhi toleransi yang diberikan. Kemudian, struktur tetap BP diperoleh.
- *Forecast*. Dimana model BP terlatih digunakan untuk perkiraan atau prediksi.

Confusion Matrix

Adalah metode umum yang biasa digunakan untuk melihat kinerja model klasifikasi yang dibuat. *Confusion matrix* juga bisa dapat dikatakan sebuah ringkasan hasil prediksi pada permasalahan klasifikasi [13]. Ada beberapa istilah yang ada pada *confusion matrix* tersebut, yaitu:

- *True Positive* (TP) : adalah ketika model berhasil memprediksi yang benar,
- *False Positive* (FP) : adalah ketika model memprediksi benar padahal salah.
- *False Negative* (FN) : adalah ketika model memprediksi salah padahal benar, dan
- *True Negative* (TN) : adalah ketika model berhasil memprediksi yang salah.

Tabel 1 Confusion Matrix

		Actual Values	
Predicted Values		True Positive (TP)	False Positive (FP)
		False Negative (FN)	True Negative (TN)

Tabel 1 merupakan bentuk dari *confusion matrix*. Lalu, adapun metrik khusus yang didapat dengan menggunakan *confusion matrix*, seperti:

Accuracy, merupakan metrik yang digunakan untuk melihat kemampuan model dalam memprediksi dengan baik, perhitungan *accuracy* dapat dilihat pada rumus dibawah ini:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision, merupakan metrik yang digunakan untuk melihat kemampuan model dalam memprediksi *True Positive* dibagi jumlah yang prediksi model yang benar dan salah, perhitungan *precision* dapat dilihat pada rumus dibawah ini:

$$Precision = \frac{TP}{TP + FP}$$

Recall, merupakan metrik yang digunakan untuk melihat kemampuan model dalam memprediksi *True Positive* yang dibagi jumlah data aktual yang *Positive*, perhitungan *recall* dapat dilihat pada rumus dibawah ini:

$$Recall = \frac{TP}{TP + FN}$$

F-Measure atau biasa disebut *F1 score*, adalah rata-rata harmonik antara *precision* dan *recall* [14], perhitungan *F1 score* dapat dilihat pada rumus dibawah ini:

$$F1\ Score = 2\ \frac{Recall\ x\ Precision}{Recall + Precision}$$

SMOTE

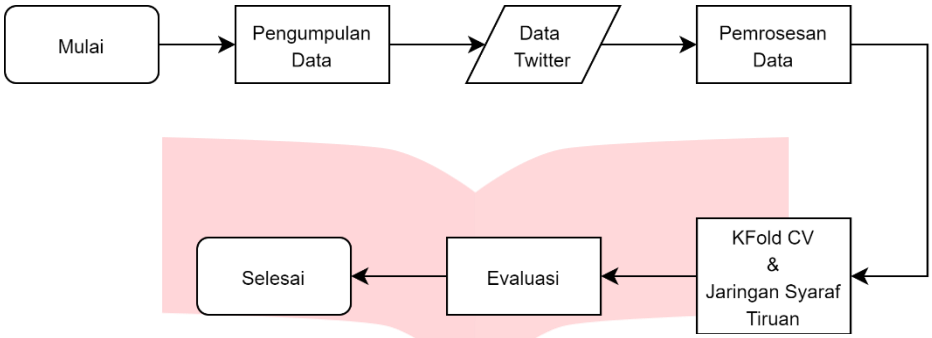
Merupakan salah satu teknik *oversampling* dengan cara menduplikasi data pada kelas minor hingga data menjadi seimbang [15]. Bekerja dengan menghasilkan *instance* baru dari kasus minoritas yang ada yang diberikan sebagai *input*. Penerapan SMOTE ini tidak mengubah jumlah kasus mayoritas. mengambil sampel ruang fitur untuk setiap kelas target dan tetangga terdekatnya, dan menghasilkan contoh baru yang menggabungkan fitur kasus target dengan fitur tetangganya [16].

NearMiss

NearMiss mengacu pada kumpulan metode *undersampling* yang memilih data berdasarkan jarak pada data kelas mayoritas ke data kelas minoritas. *NearMiss* memiliki 3 versi yaitu *NearMiss*-1, *NearMiss*-2 dan *NearMiss*-3. *NearMiss*-1 memilih data dari kelas mayoritas yang memiliki jarak rata-rata terkecil ke tiga data terdekat dari kelas minoritas [17].

3. Sistem yang Dibangun

Dalam penelitian ini, sistem yang dibangun untuk prediksi *retweet* pada data Twitter yang didapat pada 14 Juni 2021 pukul 22:29 WIB menggunakan algoritma jaringan syaraf tiruan atau *artificial neural network*. Pada Gambar 2 dapat dilihat perancangan sistem yang dibangun untuk penelitian ini:



Gambar 2 Rancangan sistem prediksi *retweet* menggunakan klasifikasi ANN

Pengumpulan Data

Data diambil pada aplikasi Twitter dengan cara *crawling* menggunakan tweepy. Data yang didapat sebanyak 11229 data serta data *tweet* yang dikumpulkan merupakan *tweet* 2 hari lalu, yaitu tweet pada tanggal 12 Juni 2021 berbahasa Indonesia dengan *query search* "covid".

Data Twitter

Data *tweet* yang didapat dari hasil *crawling* memiliki fitur berbasis pengguna dan fitur berbasi konten yang dapat dilihat pada Tabel 2. Ada sedikit perubahan pada fitur *tweet*, disini penulis menggunakan skala tahun untuk fitur age_of_account pada fitur berbasis pengguna. Lalu, penulis menggabungkan fitur contain_video dan contain_picture menjadi contain_media. Pada Tabel 3 merupakan *sample* dari *dataset* yang digunakan beserta fiturnya.

Tabel 2 Fitur Retweet yang Digunakan

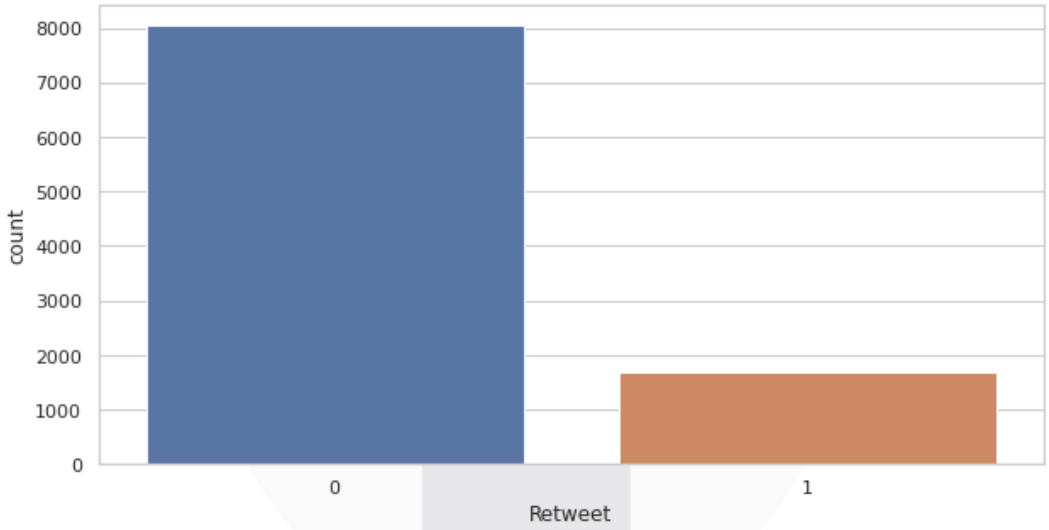
Kode Fitur	Nama Fitur	Tipe Data
F1	total_of_tweet	Numerik
F2	no_of_followers	Numerik
F3	no_of_followees	Numerik
F4	no_of_favourite	Numerik
F5	age_of_account	Numerik
F6	no_of_groups	Numerik
F7	user_name_len	Numerik
F8	contain_location	Boolean
F9	contain_tvshow	Boolean
F10	contain_media	Boolean
F11	contain_upper	Boolean
F12	contain_user_mentioned	Boolean
F13	contain_url	Boolean
F14	contain_hashtag	Boolean
F15	opt_length	Boolean
F16	len_of_text	Numerik

Tabel 3 *Sample* Data Twitter

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	Retweet
13035	295	265	885	6.58	2	12	0	0	0	0	1	0	0	0	139	1
13391	3210	1123	311	10.14	42	15	1	0	0	0	0	0	0	0	135	1
25260	11277	855	5086	10.28	18	13	1	0	0	0	0	0	0	0	140	1
5159	368	66	468	2.39	0	11	0	0	0	0	0	1	0	0	110	0
39959	2243	228	336	8.24	2	12	1	0	0	0	0	1	0	0	107	0
21807	985	3211	2283	10.85	0	11	0	0	0	0	0	0	0	0	138	1
38	34	5	2	0.11	0	11	0	0	0	0	0	0	0	0	140	1
24893	290	698	10305	8.90	3	9	0	0	0	0	1	1	0	0	106	0
6978	5509	5525	54059	0.80	0	9	0	0	0	0	1	1	0	0	138	0
4946	10347	90	28	6.22	5	14	1	0	0	0	0	0	0	0	144	1

Pemrosesan Data

Data yang didapat diolah terlebih dahulu agar siap digunakan pada model. Dengan melakukan pembersihan data dengan mengecek duplikasi pada data sehingga data yang sebelumnya 11229 data menjadi 9706 data. setelah itu, dilakukannya pengecekan terhadap *imbalance class* pada data yang diperoleh, dimana kelas 0 adalah kelas *tweet* tidak di-*retweet* sedangkan untuk kelas 1 adalah kelas *tweet* yang di-*retweet*. Data yang diperoleh memiliki ketidakseimbangan sehingga kelas 0 lebih banyak dibandingkan kelas 1. Kelas 0 memiliki data sebanyak 8034. Lalu, untuk kelas 1 memiliki data sebanyak 1672 data. Gambar 3 adalah visualisasi hasil pengecekan ketidakseimbangan kelas. Untuk proses *handling imbalance class* ini diatasi yang juga digunakan sebagai skenario pengujian yang dibagi menjadi 2 skenario yaitu melakukan *oversampling* menggunakan metode SMOTE dan *undersampling* menggunakan metode *NearMiss* pada *library* imblearn yang akan dibahas pada bagian selanjutnya.



Gambar 3 Distibusi Kelas Retweet

K-*Fold Cross Validation*

Setelah selesai melakukan pemrosesan data maka data siap digunakan. Lalu, penulis menggunakan 10-*Fold Cross Validation* sehingga pembagian datanya menjadi 90% untuk data latih dan 10% untuk data uji dari total data yang digunakan sebanyak 9706 data.

Algoritma Jaringan Syaraf Tiruan

Pada penelitian ini digunakan *hidden layer* sebanyak 2 *layer*. Lalu, fungsi aktivasi yang digunakan adalah fungsi aktivasi *logistic* atau biasa disebut *sigmoid* dan menggunakan inisiasi *learning rate* sebanyak 4 yaitu saat inisiasi *learning rate* 0.1, 0.01, 0.001 dan 0.0001 dengan jumlah neuron pada *hidden layer* adalah (32, 50) dan (128, 50) untuk masing masing inisiasi *learning rate*-nya. Parameter model ANN untuk prediksi *retweet* pada penelitian ini dapat dilihat pada Tabel 4.

Table 4 Model ANN untuk Prediksi Retweet

Model	Hidden Layer	Learning Rate Initial
MLPClassifier1	(32, 50)	0.1
MLPClassifier2	(128, 50)	0.1
MLPClassifier3	(32, 50)	0.01
MLPClassifier4	(128, 50)	0.01
MLPClassifier5	(32, 50)	0.001
MLPClassifier6	(128, 50)	0.001
MLPClassifier7	(32, 50)	0.0001
MLPClassifier8	(128, 50)	0.0001

Skenario Pengujian 1

Skenario ini adalah proses pemodelan *machine learning* menggunakan *dataset* yang belum diatasi ketidakseimbangan kelasnya. *Dataset* yang digunakan diproses terlebih dahulu dengan melakukan pembersihan data dengan mengecek duplikasinya.

Skenario Pengujian 2

Skenario ini adalah proses pemodelan *machine learning* menggunakan *dataset* yang telah dilakukan *oversampling*. Teknik *oversampling* ini menggunakan metode SMOTE pada *library* imblearn, dari 9706 data menjadi 16068 dengan masing masing data kelas 0 dan kelas 1 adalah 8034 data. Setelah *dataset* di-*oversampling*, data langsung diimplementasikan pada model *neural network* dengan parameter yang sama dengan parameter yang dilakukan di skenario pengujian 1. Dengan teknik SMOTE ini, *minority class* di duplikasi dari 1672 data sehingga sama banyak dengan *majority class* yaitu 8034 data.

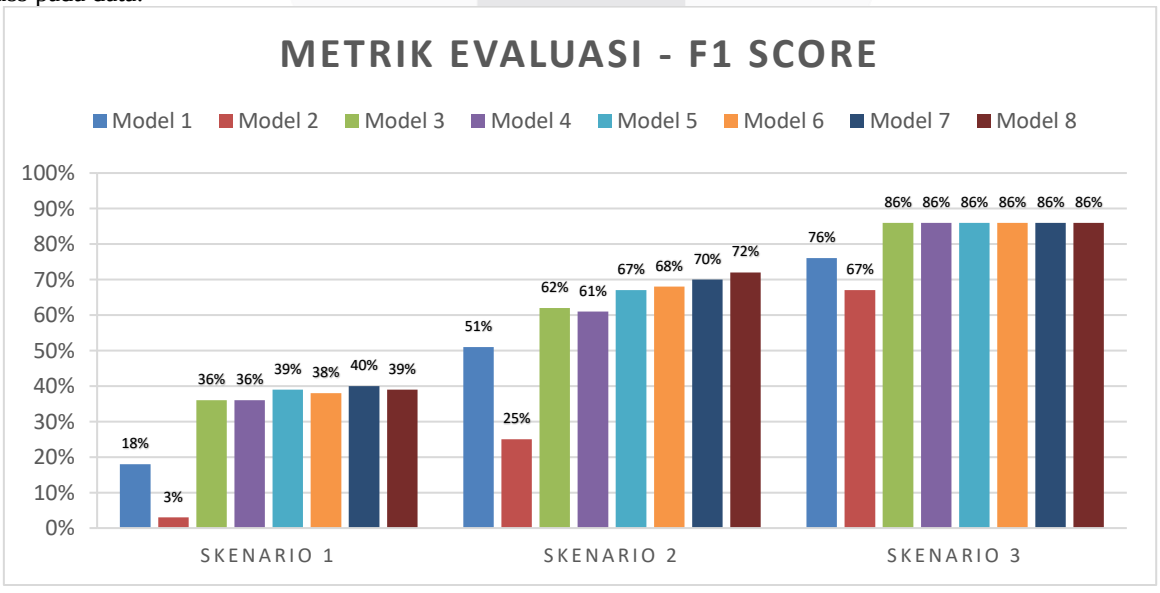
Skenario Pengujian 3

Skenario ini adalah proses pemodelan *machine learning* menggunakan *dataset* yang telah dilakukan *undersampling*. Teknik *undersampling* ini menggunakan metode *NearMiss* pada *library* imblearn, dari 9706 data menjadi 3344 dengan masing masing data kelas 0 dan kelas 1 adalah 1672. Setelah *dataset* di *undersampling*, data langsung diimplementasikan pada model *neural network* dengan parameter yang sama dengan parameter yang dilakukan di skenario pengujian 1 dan skenario pengujian 2. Dengan teknik *NearMiss* ini, dari 8034 data *majority class* menjadi sama banyak dengan *minority class* yaitu 1672 data.

4. Evaluasi

Hasil Skenario Pengujian

Gambar 4 merupakan hasil dari setiap skenario pengujian pada model prediksi *retweet* dimana metrik yang digunakan adalah metrik F1 *score*. Karena F1 *score* adalah metrik yang digunakan pada saat terjadi *imbalance class* pada data.



Gambar 4 Hasil untuk Setiap Skenario Pengujian

Analisis Hasil Pengujian

Berdasarkan hasil pengujian yang dilakukan bahwa hidde_layer_size dan penggunaan learning_rate_init mempengaruhi performansi model. Dapat dilihat pada saat model 1 ke model 2 yaitu perubahan hidden_layer_size dari (32, 50) ke (128, 50) dengan learning_rate_initial = 0.1 itu mempengaruhi kinerja model untuk setiap skenario pengujian. Lalu, pada model 3 dan model 4 dimana learning_rate_init nya diperkecil menjadi 0.01 itu meningkatkan performansi model untuk setiap skenario pengujian. Tetapi untuk skenario pengujian 3, dari model 3 sampai model 8 sudah menghasilkan performansi yang konsisten di 86%. Sedangkan untuk skenario 1 dan skenario 2, model dengan parameter yang ditentukan masih memberikan pengaruh pada kinerja model. Jika dilihat kembali pada model bernomor ganjil, perubahan hidden_layer_size itu meningkatkan performansi model ketika parameter learning_rate_init pada saat 0.01 hingga 0.0001. Lalu, untuk informasi tambahan bahwa nilai performansi meningkat drastis ketika melakukan penanganan *imbalance class*, seperti pada model 1 dibandingkan model 2 juga model 1 dibandingkan model 3. Tetapi jika dibandingkan secara keseluruhan, model 3 menghasilkan performansi yang lebih baik dibandingkan dengan model 1 dan model 2.

5. Kesimpulan

Pada penelitian ini dapat disimpulkan bahwa penggunaan fitur berbasis pengguna dan fitur berbasis konten dapat mempengaruhi terjadinya retweet. Lalu, model terbaik adalah model yang mengatasi *imbalance class* dengan metode *undersampling* menggunakan *NearMiss* dari imblearn yang menghasilkan 86% F1 *score*. Pada masa mendatang diharapkan adanya penelitian lebih lanjut untuk penelitian ini seperti menggunakan *deep learning* atau membandingkan dengan metode lain.

Referensi

[1] Daga, I., Gupta, A., Vardhan, R., & Mukherjee, P. (2020). Prediction of Likes and Retweets Using Text Information Retrieval. Procedia Computer Science, 168, 123-128.
[2] Maind, S. B., & Wankar, P. (2014). Research paper on basic of artificial neural network. International Journal on Recent and Innovation Trends in Computing and Communication, 2(1), 96-100.
[3] Suh, B., Hong, L., Pirolli, P., & Chi, E. H. (2010). Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In 2010 IEEE Second International Conference on Social Computing (pp. 177-184). IEEE.
[4] Puspitasari, D., Ramanda, K., Supriyatna, A., Wahyudi, M., Sikumbang, E. D., & Sukmana, S. H. (2020, November). Comparison of Data Mining Algorithms Using Artificial Neural Networks (ANN) and Naive Bayes for Preterm Birth Prediction. In *Journal of Physics: Conference Series* (Vol. 1641, No. 1, p. 012068). IOP Publishing.
[5] Fong, A., Sibley, C., Cole, A., Baldwin, C., & Coyne, J. (2010). A comparison of artificial neural networks, logistic regressions, and classification trees for modeling mental workload in real-time. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 54, No. 19, pp. 1709-1712). Sage CA: Los Angeles, CA: SAGE Publications.
[6] Hoang, T. B. N., & Mothe, J. (2018). Predicting information diffusion on Twitter–Analysis of predictive features. Journal of computational science, 28, 257- 264.
[7] Zhang, Q., Gong, Y., Wu, J., Huang, H., & Huang, X. (2016). Retweet prediction with attention-based deep neural network. In Proceedings of the 25th ACM international on conference on information and knowledge management (pp. 75-84).
[8] Brownlee, J. 2018. A Gentle Introduction to k-fold Cross-Validation. [Online] Available at: https://machinelearningmastery.com/k-fold-cross-validation/ [Accessed 19 June 2021]
[9] Suzuki, K. (Ed.). (2011). Artificial neural networks: methodological advances and biomedical applications. BoD–Books on Demand.
[10] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11), e00938.
[11] Hassanipour, S., Ghaem, H., Arab-Zozani, M., Seif, M., Fararouei, M., Abdzadeh, E., ... & Paydar, S. (2019). Comparison of artificial neural network and logistic regression models for prediction of outcomes in trauma patients: A systematic review and meta-analysis. Injury, 50(2), 244-250.
[12] Huang, D., & Wu, Z. (2017). Forecasting outpatient visits using empirical mode decomposition coupled with back-propagation artificial neural networks optimized by particle swarm optimization. PloS one, 12(2), e0172539.
[13] Brownlee, J. 2016. What is a Confusion Matrix in Machine Learning. [Online] Available at: https://machinelearningmastery.com/confusion-matrix-machine-learning/ [Accessed 22 June 2021].
[14] Koehrsen, W. 2018. Beyond Accuracy: Precision and Recall. [Online] Available at: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c [Accessed 22 June 2021]

[15] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.
[16] 2019. SMOTE. [Online] Available at: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/smote [Accessed 25 Juni 2021]
[17] Brownlee, J. 2020. Undersampling Algorithms for Imbalanced Classification. [Online] Available at: https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/ [Accessed 26 Juni 2021]