

SENTIMENT ANALYSIS CLASSIFICATION IN WOMEN'S E-COMMERCE REVIEWS WITH MACHINE LEARNING APPROACH

Alfiki Diastama Afan Firdaus¹, Rizki Dwi Rahmawan², Yuzzar Rizky Mahendra³, Hasan Dwi Cahyono^{*4}

^{1,2,3,4}Faculty of Technology Information and Data Science, Universitas Sebelas Maret, Indonesia

Email: ¹alfiki@student.uns.ac.id, ²rizkidwirahmawan@student.uns.ac.id,

³yuzzarrizkymahendra@student.uns.ac.id, ⁴hasandc@staff.uns.ac.id

(Article received: June 26, 2024; Revision: August 08, 2024; published: December 29, 2024)

Abstract

User reviews on e-commerce are one of the important elements in e-commerce. User reviews can help potential buyers make decisions based on the experiences and opinions of other people, for example women's e-commerce reviews. In providing positive, neutral or negative sentiment reviews, understanding customer perceptions is challenging. Classifying sentiment reviews will solve this problem, several classification techniques have been carried out, but there is still room for development in the use of simple machine learning techniques and sampling to overcome data class imbalance. Classification techniques used in this paper include Naive Bayes, SVM, and KNN. These algorithms will be compared to determine the most accurate model. Several preprocessing techniques are also carried out such to balance the dataset using ROS and SMOTE. It was obtained that the SVM method with ROS had the highest accuracy of around 0.94 for accuracy value, 0.93 for precision value, 0.94 for recall, and 0.92 for F1-score value. This research shows that the use of sampling techniques such as ROS and SMOTE can be effective in balancing imbalanced datasets, thereby improving model classification performance. These findings can be a reference for developing more efficient and accurate sentiment classification models, especially in the case of imbalanced data.

Keywords: *k*-nearest neighbour, Naive Bayes, sentiment analysis, support vector machine, user reviews.

KLASIFIKASI SENTIMEN ANALISIS PADA WOMEN'S E-COMMERCE REVIEWS DENGAN PENDEKATAN MACHINE LEARNING

Abstrak

Ulasan pengguna pada e-commerce merupakan salah satu elemen yang penting dalam e-commerce. Ulasan pengguna dapat membantu calon pembeli dalam membuat keputusan berdasarkan pengalaman dan pendapat orang lain, contohnya pada ulasan e-commerce wanita. Masalah yang sering dihadapi adalah terkadang sulitnya memahami persepsi pelanggan dalam memberikan ulasan sentimen positif, netral, maupun negatif. Mengklasifikasi ulasan sentimen ini akan menyelesaikan permasalahan tersebut, beberapa teknik klasifikasi sudah dilakukan, namun masih ada ruang pengembangan pada penggunaan teknik *machine learning* sederhana dan sampling untuk mengatasi ketidakseimbangan kelas data. Teknik klasifikasi yang digunakan antara lain Naive Bayes, SVM, dan KNN. Algoritma tersebut akan dibandingkan untuk menentukan model yang paling akurat. Beberapa teknik preprocessing juga dilakukan seperti menyeimbangkan *dataset* menggunakan ROS dan SMOTE. Diperoleh hasil metode SVM dengan ROS memiliki akurasi paling tertinggi sekitar 0.94 untuk nilai *accuracy*, 0.93 untuk nilai *precision*, 0.94 untuk *recall*, dan 0.92 untuk nilai *F1-score*. Penelitian ini menunjukkan bahwa penggunaan teknik sampling seperti ROS dan SMOTE dapat efektif dalam menyeimbangkan *dataset* yang tidak seimbang, sehingga meningkatkan performa model klasifikasi. Temuan ini dapat menjadi acuan untuk pengembangan model klasifikasi sentimen yang lebih efisien dan akurat, khususnya dalam kasus data yang tidak seimbang.

Kata kunci: *analisis sentimen, k*-nearest neighbour, Naive Bayes, review pengguna, support vector machine

1. PENDAHULUAN

E-commerce telah menjadi salah satu sektor yang berkembang pesat dengan berbagai platform yang tersedia. Hal ini telah mengubah cara konsumen

berbelanja dan berinteraksi dengan produk dan layanan [1]. Salah satu elemen yang penting dalam e-commerce adalah ulasan pengguna. Ulasan pengguna dapat membantu calon pembeli dalam membuat

keputusan lebih baik berdasarkan pengalaman dan pendapat orang lain [2]. Dalam hal ini, ulasan *e-commerce* wanita memiliki peran penting karena sering kali wanita menjadi pengambil keputusan penting dalam hal belanja kebutuhan pakaian. [3]

Menganalisis ulasan pelanggan membantu konsumen dalam memahami sebuah produk atau layanan sebelum membuat keputusan untuk membeli. Ulasan pelanggan ini juga bermanfaat bagi penyedia layanan dalam meningkatkan kualitas layanan dan memberi pemahaman yang lebih baik tentang bagaimana mereka mungkin lebih lemah dari pesaing [4].

Salah satu analisis yang dapat dilakukan pada ulasan pengguna adalah analisis sentimen. Analisis sentimen merupakan penilaian, tanggapan, serta perasaan yang dihasilkan dari teks oleh algoritma kecerdasan buatan yang bergantung pada teknik *machine learning*. Penilaiannya terdiri dari positif dan negatif [5]. *Natural Language Processing* (NLP) sangat terlibat dengan analisis sentimen yang biasanya digunakan untuk memahami suara pelanggan dari ulasan, tanggapan survei, media sosial, dan bidang seperti pemasaran untuk meningkatkan *customer service* [6].

Melalui analisis sentimen, pembentukan keputusan dibuat berdasarkan data yang signifikan daripada intuisi yang sederhana [7]. Hasil keputusan yang berdasarkan data dapat menciptakan hasil keputusan yang lebih objektif, dengan harapan perubahan dan keputusan yang diambil memang dibutuhkan di kejadian nyata. Namun, hal yang sulit ketika bekerja dengan analisis sentimen adalah sarkasme, ketergantungan konteks, bahasa gaul, singkatan, data *multilingual*, dan lintas domain. Hal tersebut mempersulit proses ekstraksi fitur teks [8].

Beberapa pendekatan telah dilakukan untuk melakukan analisis sentimen terhadap ulasan *e-commerce*. Studi pertama [1] menggunakan *bidirectional recurrent neural network* (RNN) dengan *long-short term memory* (LSTM) menghasilkan *F1-score* 0.88 untuk klasifikasi rekomendasi dan 0.93 untuk klasifikasi sentimen. Studi selanjutnya [3] menggunakan pendekatan klasifikasi Naive Bayes, AdaBoost, JRip, J48, dan Sequential Minimal Optimization (SMO), menghasilkan performa terbaik oleh SMO dengan nilai akurasi 80.875% karena hampir semua masalah pengkategorian teks dapat dipisahkan secara linier. Dari hasil penelitian yang telah dilakukan sebelumnya, terdapat beberapa ruang pengembangan yang dapat dilakukan seperti *preprocessing* data, ketidakseimbangan data, dan penggunaan metode klasifikasi lainnya.

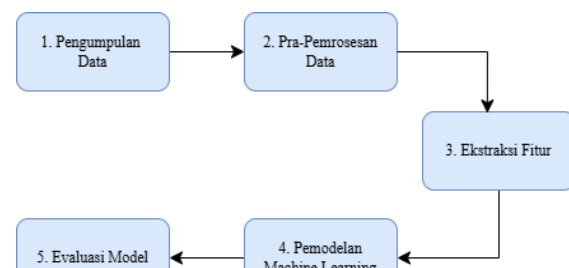
Ketidakseimbangan data menjadi sebuah permasalahan yang perlu diperhatikan karena ketika terjadi ketidakseimbangan kelas dalam data pelatihan, model biasanya akan mengklasifikasikan kelompok mayoritas secara berlebihan karena peningkatan probabilitas sebelumnya. Akibatnya,

kasus-kasus yang termasuk dalam kelompok minoritas lebih sering mengalami kesalahan klasifikasi dibandingkan kasus-kasus yang termasuk dalam kelompok minoritas kelompok mayoritas [9].

Studi ini bertujuan untuk menganalisis sentimen dari Women's Clothing E-Commerce Reviews [10] dengan metode *machine learning* Support Vector Machine (SVM), Naive Bayes, dan K-Nearest Neighbour (KNN). Menguji metode *machine learning* KNN karena metode berikut menyimpan *data value set* dan mengklasifikasikan *data value set* baru dengan pengukuran yang serupa [3]. Selain itu, tahapan *preprocessing data* juga menggunakan *tf-idf* dan untuk mengatasi ketidakseimbangan data, digunakan beberapa teknik *sampling* seperti random oversampling (ROS) dan SMOTE.

2. METODE PENELITIAN

Dalam mengolah teks ulasan dan menggunakannya untuk analisis sentimen diperlukan beberapa proses yang dilakukan.



Gambar 1. Metode Penelitian untuk *sentiment analysis* pada *Women E-Commerce reviews*

Metode yang digunakan untuk melakukan penelitian *sentiment analysis* pada *Women E-Commerce reviews* ini memiliki beberapa tahapan seperti yang ditunjukkan pada Gambar 1. Tahapan yang digunakan yaitu: (1) pengumpulan data, (2) pra-pemrosesan data, (3) ekstraksi fitur, (4) pemodelan *machine learning*, (5) evaluasi model. Setiap tahapan ini memiliki kontribusi pada keseluruhan proses analisis sentimen untuk menghasilkan model yang akurat dan dapat diandalkan.

2.1. Pengumpulan Data

Penelitian ini menggunakan *dataset* dari Kaggle “*Women's E-Commerce Clothing Reviews*”. *Dataset* ini mencakup berbagai ulasan produk yang diberikan oleh konsumen, baik itu ulasan positif maupun negatif. Ada 10 atribut di *dataset* ini dan hanya ada 1 atribut yang digunakan untuk *sentiment analysis*. Atribut yang dipakai yaitu “*Review Text*” yang ditunjukkan dalam Tabel 1. berisi semua ulasan, komentar, dan opini dari konsumen kepada suatu produk.

2.2. Preprocessing Data

Pra-pemrosesan data adalah tahapan yang krusial. Tujuan dari tahapan ini adalah untuk membersihkan *dataset* dari data yang tidak berguna dan mengganggu sehingga dapat dihasilkan penemuan yang lebih akurat [11]. Tabel 1 menunjukkan contoh *review text* yang ada pada *dataset* yang digunakan. Pra-pemrosesan data ini akan dilakukan langkah-langkah yaitu, pertama pembersihan teks ulasan dengan menghapus beberapa delimiter yang ditemukan dalam teks.

Tabel 1. *Review text* dari *dataset*

Indeks	Review Text
0	Absolutely wonderful - silky and sexy and comf...
1	Love this dress! it's sooo pretty. i happene...
2	I had such high hopes for this dress and reall...
3	I love, love, love this jumpsuit. It's fun, fl...
4	This shirt is very flattering to all due to.
5	I love tracy reese dresses, but this one is no...
6	I aded this in my basket at hte last mintue to...
7	I ordered this in carbon for store pick up, an...
8	i love this dress. i usually get an xs but it ...
9	i'm 5"5' and 125 lbs. i ordered the s petite t...

Selanjutnya, ekstraksi fitur dilakukan agar teks dapat diklasifikasikan, termasuk penghapusan stop words untuk menghilangkan kata-kata umum seperti "at" dan "is", mengubah seluruh teks menjadi huruf kecil (lowercase) untuk memastikan konsistensi dan menghindari duplikasi karena perbedaan huruf besar dan kecil, serta menggunakan tokenizer untuk memecahkan string besar menjadi daftar kata-kata dengan bantuan library Natural Language Toolkit (NLTK). Tokenizer digunakan untuk *tokenization* dimana kalimat dipecah menjadi unit bermakna yang lebih kecil dinamakan token [12]. Beberapa teknik pemrosesan awal lainnya yang digunakan adalah:

2.2.1. Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency atau TF-IDF adalah teknik yang digunakan untuk mengukur seberapa penting suatu kata dalam sebuah dokumen, relatif terhadap kumpulan dokumen lainnya. Proses ini melibatkan dua langkah utama mulai dari menghitung Term Frequency (TF) dan Inverse Document Frequency (IDF). TF mengukur seberapa sering sebuah kata muncul dalam dokumen. Jika $n(t, d)$ adalah frekuensi kemunculan kata t dalam dokumen d , maka:

$$TF(t, d) = \frac{n(t, d)}{\sum_k(k, d)} \quad (1)$$

Sedangkan, IDF mengukur seberapa penting sebuah kata dalam seluruh korpus dokumen. Kata dengan nilai IDF yang tinggi dianggap penting karena kata tersebut jarang muncul di dokumen lainnya [13]. Sedangkan, N adalah total jumlah dokumen dan $DF(t)$ adalah jumlah dokumen yang mengandung kata t , maka:

$$IDF(t) = \log\left(\frac{N}{DF(t)}\right) \quad (2)$$

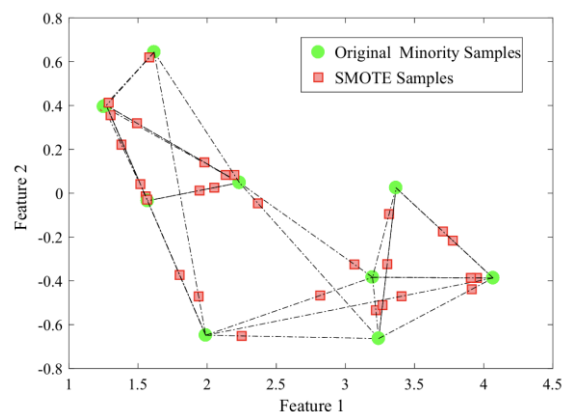
Selanjutnya, TF-IDF didapatkan dari perkalian produk dari TF dan IDF maka:

$$TFIDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

Penggunaan TF-IDF ini akan membuat setiap ulasan teks diubah menjadi vektor fitur yang merepresentasikan kata-kata dalam konteks keseluruhan dokumen ulasan. TF-IDF banyak digunakan untuk *classification*, *clustering*, dan *visualization*.

2.2.2. Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) adalah metode *oversampling* yang ide utamanya adalah membentuk contoh kelas minoritas baru dengan mengambil sampel secara acak segmen garis antara instans minoritas tetangga. [14] Memperbesar ukuran data seharusnya meningkatkan performa klasifikasi dan mengurangi dampak buruk efek sampel yang kecil. Dengan menambahkan data sintetik, model dapat belajar dari variasi yang lebih luas dalam data minoritas, sehingga memperbaiki kemampuan generalisasi dan akurasi klasifikasi [15]. Gambar 2 menunjukkan mekanisme pembentukan data sintetik menggunakan SMOTE.



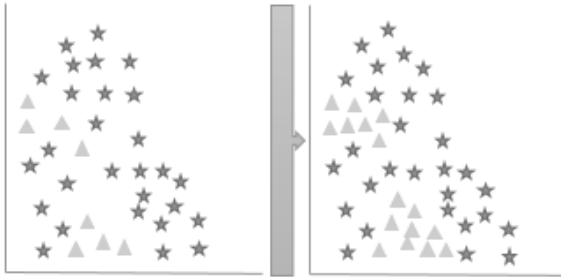
Gambar 2. Contoh pembuatan sampel minoritas menggunakan mekanisme interpolasi SMOTE

Dapat dipahami bahwa pola SMOTE berada pada garis koneksi antar sampel kelas minoritas dan K-Nearest Neighbour (KNN) [16].

2.2.3. Random Oversampling (ROS)

Random Oversampling atau ROS merupakan metode *oversampling* dimana data poin dari kelas minoritas dipilih secara acak dan diduplikasikan secara persis. Tujuannya adalah untuk menambah jumlah sampel pada kelas minoritas sehingga menciptakan keseimbangan antara dua kelas. [17]. Namun, peningkatan ukuran kumpulan data dengan ROS juga berpotensi meningkatkan biaya komputasi [18]. Selain itu, pendekatan ini juga berpotensi

menciptakan *overfitting* dikarenakan penduplikasian *instance* yang terlalu banyak. [19]



Gambar 3. Contoh dari Random Oversampling

Gambar 3 menunjukkan gambaran dari *oversampling* dimana bagian sebelah kiri merupakan data sebelum random over sampling dan sebelah kanan kondisi setelah penambahan data kelas minoritas dengan random oversampling.

2.3. Pemodelan Machine Learning

2.3.1. Naive Bayes

Naive Bayes [20] adalah algoritma pembelajaran sederhana yang menggunakan Teorema Bayes yang diformulasikan dengan:

$$P(y | x) = P(y)P(X | y)/P(x) \quad (4)$$

bersama dengan asumsi bahwa setiap atribut independen secara kondisional,

$$P(x | y) = \prod_{i=1}^n P(x_i | y) \quad (5)$$

dimana x_i merupakan nilai atribut ke- i dalam x dan n adalah jumlah dari atribut. Selanjutnya,

$$P(x) = \prod_{i=1}^k P(c_i) P(x | c_i) \quad (6)$$

dimana k adalah jumlah dari kelas dan c_i adalah kelas ke- i . Walaupun *Naive Bayes classifier* sederhana, pengklasifikasi ini efektif karena ketahanannya terhadap fitur yang tidak relevan. Memiliki kinerja yang baik di domain dengan banyak fitur yang sama pentingnya. Akurasi dari pengklasifikasi ini meningkat dengan adanya tahapan pra-pemrosesan data untuk menghilangkan *noise* [21].

2.3.2. Support Vector Machine (SVM)

Model SVM menggunakan kernel linear, yang didefinisikan oleh fungsi berikut.

$$f(x) = w \cdot x + b \quad (7)$$

Dengan w adalah vektor bobot, x adalah vektor fitur dari data input, dan b adalah bias. SVM yang digunakan diharapkan dapat menemukan *hyperplane* yang memisahkan kelas positif dan negatif dengan margin terbesar. Margin adalah jarak terpendek dari titik data terdekat dari titik data terdekat (*support vectors*) ke *hyperplane*. Tujuannya adalah

memaksimalkan margin, yang secara matematis dinotasikan sebagai berikut.

$$margin = \frac{2}{||w||} \quad (8)$$

Untuk menemukan *hyperplane* yang optimal, SVM meminimalkan fungsi *loss* yang bertujuan pada dua hal, yaitu meminimalkan norma dari vektor bobot ($||w||$) dan meminimalkan kesalahan pada klasifikasi. Fungsi *loss* yang digunakan adalah sebagai berikut.

$$L(w, b, \alpha) = \frac{1}{2} ||w||^2 + C \sum_{i=1}^n \max(0, 1 - y_i(w \cdot x_i + b)) \quad (9)$$

$||w||$ adalah norma dari vektor bobot yang mengontrol margin, C adalah parameter regularisasi yang mengontrol *trade-off* antara margin yang besar dan kesalahan klasifikasi. y_i adalah label kelas (1 untuk positif, dan -1 untuk negatif). x_i adalah vektor fitur dari titik data ke- i .

Parameter C adalah *hyperparameter* yang harus ditentukan sebelumnya dan ini berpengaruh pada performa model. Nilai C yang lebih besar akan membuat model lebih menekankan pada kesalahan klasifikasi. Sebaliknya, nilai C yang lebih kecil akan menekankan pada margin yang lebih besar, sehingga model akan lebih toleran terhadap kesalahan klasifikasi tetapi menghasilkan batas pemisah yang lebih sederhana dan umum. Pemilihan nilai C yang tepat sangat penting untuk mencapai keseimbangan antara kompleksitas model dan kemampuannya untuk menggeneralisasi data baru.

2.3.3. K-Nearest Neighbour (KNN)

KNN merupakan salah satu algoritma yang paling efektif untuk segregasi data ketika data yang diberikan cukup ambigu [22]. KNN digunakan untuk klasifikasi dan regresi dengan cara mencari sejumlah data tetangga terdekat (*neighbors*) yang paling mirip dengan data baru untuk melakukan prediksi atau klasifikasi [20]. Algoritma KNN bekerja berdasarkan prinsip kesamaan, di mana data baru diklasifikasikan berdasarkan mayoritas kelas dari tetangga-tetangga terdekatnya. Proses kerjanya melibatkan beberapa langkah utama:

1. Pemilihan K

Pertama kita menentukan jumlah tetangga terdekat (K) yang akan digunakan. Nilai K ini akan mempengaruhi hasil prediksi model. Pemilihan K yang tepat sangat penting untuk kinerja algoritma KNN.

2. Perhitungan Jarak

Menghitung jarak antara data baru dengan seluruh data yang ada dalam *dataset*. Jarak yang paling sering digunakan adalah jarak Euclidean. Jarak ini dihitung menggunakan rumus berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (10)$$

di mana p dan q adalah dua titik data, dan x_i dan y_i adalah koordinat ke-i dan titik x dan y.

3. Pemilihan Tetangga Terdekat

Memilih K data terdekat berdasarkan jarak yang telah dihitung. Nilai K dalam metode KNN ini sangat penting karena menentukan keakuratan dan efektivitas algoritma [20]. Pemilihan nilai K yang tepat sangat bergantung pada *dataset* dan masalah yang dihadapi. Jika K terlalu kecil, algoritma KNN menjadi sangat sensitif terhadap *noise* dalam data, yang dapat menyebabkan *overfitting*. Sebaliknya, jika K terlalu besar, algoritma dapat menjadi terlalu umum dan kehilangan detail penting, yang dapat menyebabkan *underfitting*.

2.4. Evaluasi Model

Evaluasi model dilakukan dengan menggunakan beberapa metrik kinerja untuk menilai seberapa baik model dalam mengklasifikasikan ulasan *woman e-commerce*. Metrik-metrik yang digunakan meliputi akurasi, *confusion matrix*, dan laporan klasifikasi, yang berisi *precision*, *recall*, dan *F1-Score*. Akurasi dihitung melalui persamaan:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

- TP adalah *True Positives* (prediksi positif yang benar).
- TN adalah *True Negatives* (prediksi negatif yang benar).
- FP adalah *False Positives* (prediksi positif yang salah).
- FN adalah *False Negatives* (prediksi negatif yang salah).

Confusion Matrix merupakan matriks yang menunjukkan performa model dengan merinci jumlah prediksi benar dan salah untuk setiap kelas. Matriks ini membantu untuk memahami kesalahan yang dibuat oleh model dan distribusi dari prediksi yang salah.

Precision merupakan rasio antara prediksi positif yang benar dengan keseluruhan data yang diprediksi positif oleh model. *Precision* diformulasikan dengan:

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

Recall merupakan rasio antara prediksi positif yang benar dari model dengan keseluruhan data yang nilainya positif. *Recall* diformulasikan dengan:

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

F1-Score merupakan *harmonic mean* dari *precision* dan *recall*. Metrik ini digunakan untuk memudahkan proses perbandingan antara metode. *F1-Score* diformulasikan dengan:

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (14)$$

3. HASIL DAN PEMBAHASAN

3.1. Pra-pemrosesan Data dan Ekstraksi Fitur

Pada pra-pemrosesan data, data akan dilakukan pembersihan data seperti kata-kata *stopwords* yang ada di *review text*. Hal ini digunakan untuk menghapus kata-kata yang kurang berguna dalam sentimen analisis. Dengan menghilangkan stopwords, model dapat fokus pada kata-kata yang lebih relevan dan signifikan untuk penilaian sentimen, sehingga meningkatkan akurasi analisis. Selanjutnya, kita akan melihat contoh kode yang digunakan untuk mengidentifikasi dan menghapus stopwords dari data teks.

Selanjutnya akan dilakukan perubahan huruf besar menjadi huruf kecil agar menjaga konsistensi huruf dan menghindari duplikasi yang diakibatkan perbedaan huruf besar dan kecil. Dengan mengubah semua teks menjadi huruf kecil, kita memastikan bahwa kata-kata yang sama, meskipun ditulis dengan variasi huruf besar dan kecil, dianggap seragam dan tidak terhitung lebih dari sekali. Proses ini penting untuk mengurangi kecacauan data dan meningkatkan kualitas analisis sentimen, karena model dapat lebih mudah memproses dan mengidentifikasi pola dalam teks yang telah distandarisasi.

Pra-pemrosesan yang terakhir adalah melakukan *tokenizer*, yaitu proses yang penting dalam menyiapkan data untuk analisis lebih lanjut. *Tokenizer* berfungsi untuk memecah teks menjadi unit-unit dasar, seperti kata atau frasa, sehingga data teks dapat diproses secara lebih terstruktur. Dengan menggunakan *tokenizer*, kita dapat mengidentifikasi dan memisahkan elemen-elemen penting dari teks, seperti kata-kata, tanda baca, dan simbol, yang memungkinkan sistem untuk memahami dan menganalisis teks dengan lebih baik. Kode yang digunakan akan saat pra-pemrosesan dapat dilihat pada Gambar 4.

```
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import RegexpTokenizer

ps = PorterStemmer()

tokenizer = RegexpTokenizer(r'\w+')
stop_words = set(stopwords.words('english'))

def preprocessing(data):
    txt = data.str.lower().str.cat(sep=' ') #1
    words = tokenizer.tokenize(txt) #2
    words = [w for w in words if not w in stop_words] #3
    #words = [ps.stem(w) for w in words] #4
    return words
```

Gambar 4. Implementasi dari pra-pemrosesan data

3.2. Hasil Implementasi Naive Bayes

Pengujian algoritma Naive Bayes (NB) menggunakan layanan google colab, dengan bahasa python dan library `sklearn.naive_bayes class`

MultinomialNB yang sebelumnya telah diimpor ke dalam platform Google Colab. Penggunaan class tersebut diawali dengan pendeklarasian objek classifier dengan kelas MultinomialNB() pada Gambar 5. Kemudian, model ditrain menggunakan data *training* dengan perintah classifier.fit, barulah kemudian diakhiri dengan memprediksi hasil dengan data test menggunakan perintah classifier.predict.

```
# Initialize and train the Multinomial Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train_tfidf, y_train)

# Predict on the test set
y_pred = classifier.predict(X_test_tfidf)
```

Gambar 5. Implementasi Algoritma Naive Bayes

Kemudian, hasil dievaluasi dengan melihat *confusion matrix* dan laporan klasifikasi. Dari *confusion matrix* pada Gambar 5, dapat diinterpretasikan sebagai berikut:

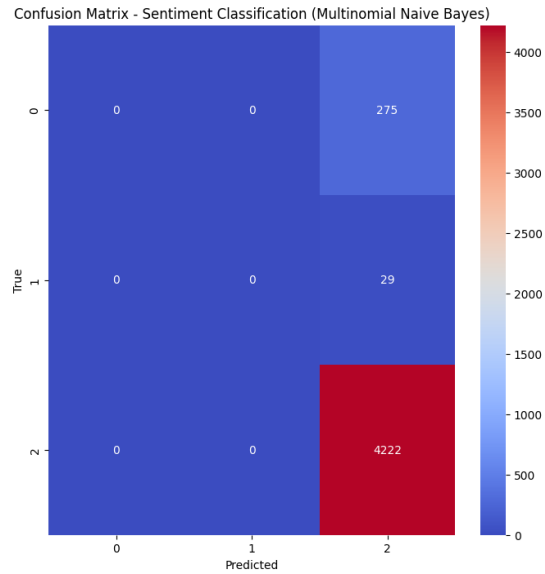
- True Positive (TP) untuk label "Negative" adalah 0.
- False Negative (FN) untuk label "Negative" adalah 275 (terklasifikasi sebagai "Positive").
- True Positive (TP) untuk label "Neutral" adalah 0.
- False Negative (FN) untuk label "Neutral" adalah 29 (terklasifikasi sebagai "Positive").
- True Positive (TP) untuk label "Positive" adalah 4222.
- False Positive (FP) untuk label "Positive" adalah 0 (terklasifikasi sebagai "Negative").

Selain itu, *confusion matrix* juga menunjukkan bahwa adanya data yang tidak seimbang dengan data positif menjadi kelas mayoritas terbanyak sejumlah 4222 data. Tabel 2 merupakan hasil evaluasi model NB biasa, tabel ini menunjukkan walaupun memiliki nilai akurasi yang tinggi 0.93, nilai *recall* untuk kelas positive 1.00 menunjukkan bahwa terdapat potensi bias terhadap kelas tersebut. Hal ini menjadi masalah karena bias yang disebabkan ketidakseimbangan data dapat menghasilkan keputusan yang salah pada kelas minoritas seperti yang ditampilkan pada Gambar 6. Sehingga untuk mengatasi ketidakseimbangan data dan potensi bias, teknik ROS dan SMOTE ditambahkan.

Hasil evaluasi pada Tabel 3 dan 4 menunjukkan bahwa penambahan teknik SMOTE mampu meningkatkan nilai *precision* metode NB dengan kenaikan 0.06 menjadi 0.93 namun mengalami penurunan untuk *accuracy*, *recall*, dan *F1-Score*, sedangkan penggunaan RS tidak menghasilkan perubahan signifikan yang dapat terlihat. Secara keseluruhan, penggunaan metode SMOTE dapat menjadi alternatif teknik untuk mengatasi data yang tidak seimbang jika ingin meningkatkan nilai dari *precision* pada metode naive bayes.

Gambar 6 berikut menunjukkan *confusion matrix* dari implementasi algoritma Naive Bayes yang mengilustrasikan performa model dalam

mengklasifikasikan data ke dalam kategori Negative (0), Neutral (1), dan Positive (2).



Gambar 6. Confusion Matrix Implementasi Naive Bayes

Tabel 2 menyajikan hasil evaluasi untuk Naive Bayes tanpa teknik oversampling, di mana nilai *precision* untuk kelas Positive adalah 0.93 dengan *accuracy* keseluruhan sebesar 0.93.

Tabel 2. Hasil Evaluasi NB

Model	Precision	Recall	F1-Score
Negative	0.00	0.00	0.00
Neutral	0.00	0.00	0.00
Positive	0.93	1.00	0.97
accuracy			0.93
macro avg	0.31	0.33	0.32
weighted avg	0.87	0.93	0.90

Tabel 3 menunjukkan hasil evaluasi Naive Bayes setelah menerapkan teknik SMOTE, di mana *precision* untuk kelas Positive meningkat menjadi 0.98, namun *accuracy* turun menjadi 0.82.

Tabel 3. Hasil Evaluasi NB SMOTE

Model	Precision	Recall	F1-Score
Negative	0.22	0.74	0.34
Neutral	0.13	0.10	0.12
Positive	0.98	0.83	0.90
accuracy			0.82
macro avg	0.44	0.56	0.45
weighted avg	0.93	0.82	0.86

Tabel 4 menunjukkan hasil evaluasi Naive Bayes dengan Random Oversampling (ROS), di mana *precision* untuk kelas Positive tetap tinggi di 0.93 dengan *accuracy* keseluruhan sebesar 0.93.

Tabel 4. Hasil Evaluasi NB ROS

Model	Precision	Recall	F1-Score
Negative	0.00	0.00	0.00
Neutral	0.15	0.17	0.16
Positive	0.93	0.99	0.96
accuracy			0.93
macro avg	0.36	0.39	0.37
weighted avg	0.87	0.93	0.90

3.3. Hasil Implementasi SVM

Model SVM dilatih dengan data *train* dan mengevaluasi performanya menggunakan data *test*. Pengujian algoritma SVM dilakukan dengan layanan Google Colab. Bahasa pemrograman yang digunakan adalah Python dengan pustaka `sklearn.svm` yang sebelumnya telah diimpor ke dalam platform Google Colab.

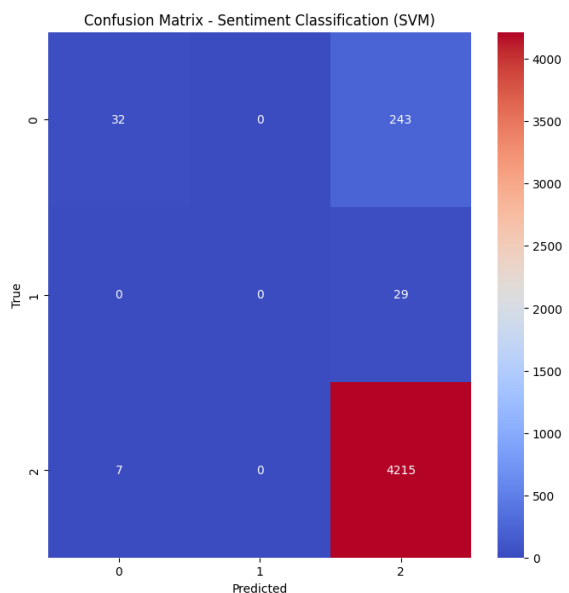
Penggunaan *class* tersebut diawali dengan pendeklarasian objek *classifier* dengan kelas `SVC()` pada Gambar 7. Kemudian, model di-*train* menggunakan *data training* menggunakan perintah `svm_classifier.fit`, barulah kemudian diakhiri dengan memprediksi hasil dengan data *test* menggunakan perintah `svm_classifier.predict`. Kemudian, hasil dievaluasi dengan melihat *confusion matrix* dan laporan klasifikasi.

```
# Initialize and train the SVM classifier
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train_tfidf, y_train)

# Predict on the test set
y_pred = svm_classifier.predict(X_test_tfidf)
```

Gambar 7. Implementasi Algoritma SVM

Berikut adalah hasil evaluasi model SVM tanpa menggunakan SMOTE dan ROS yang ditunjukkan melalui *confusion matrix*



Gambar 8. *Confusion matrix* hasil klasifikasi sentimen menggunakan SVM

Dari *confusion matrix* pada Gambar 8, dapat diinterpretasikan sebagai berikut:

- True Positive (TP) untuk label "Negative" adalah 32.
- False Negative (FN) untuk label "Negative" adalah 243 (terklasifikasi sebagai "Positive").
- True Positive (TP) untuk label "Neutral" adalah 0.

- False Negative (FN) untuk label "Neutral" adalah 29 (terklasifikasi sebagai "Positive").
- True Positive (TP) untuk label "Positive" adalah 4215.
- False Positive (FP) untuk label "Positive" adalah 7 (terklasifikasi sebagai "Negative").

Laporan klasifikasi yang diperoleh sebagai hasil evaluasi dapat dilihat pada Tabel 5. Dari hasil yang diperoleh, beberapa poin penting dapat diambil untuk dievaluasi: Akurasi keseluruhan dari model adalah 94%, yang menunjukkan bahwa model SVM dengan kernel linear ini memiliki kinerja yang baik dalam mengklasifikasikan sentimen ulasan *e-commerce* secara umum. Kelas "Positive" memiliki performa terbaik dengan *precision* sebesar 0.94, *recall* sebesar 1.00, dan *F1-score* sebesar 0.97. Hasil performa kelas *positive* ini menunjukkan bahwa model sangat baik dalam mendeteksi ulasan positif.

Tabel 5. Hasil Evaluasi SVM

Model	Precision	Recall	F1-Score
Negative	0.82	0.12	0.20
Neutral	0.00	0.00	0.00
Positive	0.94	1.00	0.97
accuracy			0.94
macro avg	0.59	0.37	0.39
weighted avg	0.93	0.94	0.92

Kelas "Negative" memiliki *precision* yang cukup baik yaitu 0.82, tetapi *recall*-nya sangat rendah yaitu 0.12. Hal ini menunjukkan bahwa banyak ulasan negatif yang salah diklasifikasikan sebagai ulasan positif. Kelas "Neutral" memiliki performa yang sangat buruk dengan *precision*, *recall*, dan *F1-score* semuanya 0.00. Ini mengindikasikan bahwa model hampir tidak mampu mendeteksi ulasan netral sama sekali.

Salah satu faktor yang kemungkinan besar mempengaruhi performa model adalah ketidakseimbangan data (*imbalanced data*). Jumlah ulasan positif sangat dominan (4222) dibandingkan dengan ulasan negatif (275) dan netral (29). Ketidakseimbangan ini dapat menyebabkan model lebih cenderung memprediksi kelas mayoritas (positif) dan mengabaikan kelas minoritas (negatif dan netral).

Untuk mengatasi ketidakseimbangan data, digunakan teknik *oversampling* menggunakan SMOTE. SMOTE digunakan untuk menyeimbangkan jumlah sampel pada setiap kelas dalam data *train* dengan membuat sampel sintesis untuk kelas minoritas. Data *train* yang telah di-vektorisasi. Selanjutnya dilakukan *resampling* menggunakan SMOTE. Penggunaan SMOTE ini bertujuan agar setiap kelas memiliki jumlah sampel yang seimbang.

Model SVM dengan kernel linear di-*train* menggunakan data *train* yang telah di-*resampling*. Parameter `class_weight='balanced'` ditambahkan untuk menangani ketidakseimbangan dalam set data *test*. Berikut adalah hasil laporan klasifikasi setelah penambahan SMOTE:

Tabel 6. Hasil Evaluasi SVM dengan SMOTE

Model	Precision	Recall	F1-Score
Negative	0.31	0.66	0.43
Neutral	0.46	0.21	0.29
Positive	0.97	0.91	0.94
accuracy			0.89
macro avg	0.58	0.59	0.55
weighted avg	0.93	0.89	0.90

Dapat dilihat pada Tabel 6, akurasi keseluruhan sedikit menurun menjadi 89%. Namun, akurasi ini diimbangi dengan peningkatan performa pada kelas minoritas. Pada ulasan negative, Precision meningkat menjadi 0.31, recall meningkat signifikan menjadi 0.66, dan *F1-score* meningkat menjadi 0.43. Ini menunjukkan bahwa model lebih baik dalam mendeteksi ulasan negatif.

Pada ulasan neutral, *precision* meningkat menjadi 0.46, *recall* menjadi 0.21, dan *F1-score* menjadi 0.29. Meskipun masih rendah, terdapat peningkatan yang signifikan dibandingkan sebelumnya. Pada ulasan positive, *precision* sedikit menurun menjadi 0.97, *recall* menurun menjadi 0.91, dan *F1-score* menjadi 0.94. Penurunan ini diimbangi dengan peningkatan kinerja pada kelas minoritas.

Macro Avg: *precision*, *recall*, dan *F1-score* rata-rata secara makro menunjukkan peningkatan performa pada semua kelas, dengan *precision* 0.58, *recall* 0.59, dan *F1-score* 0.55. Hal ini menunjukkan model tetap mempertahankan performa keseluruhan yang baik sambil meningkatkan deteksi pada kelas minoritas. Selain menggunakan SMOTE, penelitian ini juga mengaplikasikan SVM dengan ROS. Hasil dari percobaan dapat dilihat pada Tabel 7.

Tabel 7. Hasil Evaluasi SVM dengan ROS

Model	Precision	Recall	F1-Score
Negative	0.82	0.11	0.20
Neutral	0.38	0.21	0.27
Positive	0.94	1.00	0.97
accuracy			0.94
macro avg	0.71	0.44	0.48
weighted avg	0.93	0.94	0.92

Terlihat pada tabel 7, meskipun model lebih akurat dalam prediksi yang dibuat (tingginya *precision*), model masih gagal mendeteksi sebagian besar ulasan negatif (rendahnya *recall*). Hal ini mungkin disebabkan oleh ketidakseimbangan yang sangat besar antara ulasan positif dan negatif, di mana ROS belum sepenuhnya mengatasi ketidakseimbangan ini. Peningkatan dalam *precision* dan *recall* untuk kelas netral menunjukkan bahwa ROS membantu model untuk lebih baik dalam mendeteksi ulasan netral.

Performa yang masih di bawah standar, menunjukkan bahwa kelas netral masih sulit untuk dideteksi oleh model. Performa untuk kelas positif tetap sangat baik, dengan hanya sedikit penurunan dalam *precision*. Ini menunjukkan bahwa penanganan ketidakseimbangan data dengan ROS tidak banyak mengganggu kemampuan model untuk mendeteksi ulasan positif.

3.4. Hasil Implementasi KNN

Pada analisis klasifikasi sentimen, penerapan metode KNN dengan dan tanpa ROS memberikan hasil yang sangat berbeda pada *confusion matrix*-nya. Dengan penambahan sampel dari ROS, model KNN ini mendapatkan lebih banyak data untuk mempelajari karakteristik kelas minoritas. Dalam kasus tanpa ROS, model KNN cenderung mengalami kesulitan pada pengklasifikasi sentimen dari kelas minoritas, contohnya pada kasus ini adalah pada kelas negative dan neutral. Ketidakseimbangan data menyebabkan model lebih kerap menganggap sampel dari kelas minoritas sebagai bagian dari kelas mayoritas, yang mengakibatkan tingginya tingkat kesalahan klasifikasi.

Untuk nilai akurasi sendiri, saat menggunakan ROS didapatkan hasil akurasi 0.851. Lalu nilai akurasi tanpa menggunakan ROS dan SMOTE didapatkan hasil sebesar 0.923. Kemudian yang terakhir nilai akurasi menggunakan SMOTE hanya didapatkan sebesar 0.115. Jika hanya dilihat dari nilai akurasinya saja, model KNN tanpa menggunakan ROS dan SMOTE akan unggul dibandingkan dengan model lainnya. Tetapi jika dilihat lebih lanjut lagi dari *confusion matrix* pada Gambar 8, *precision*, *recall*, dan *F1-score* pada Tabel 8, 9, dan 10, didapatkan bahwa model dengan ROS lebih unggul dibandingkan dengan yang lain.

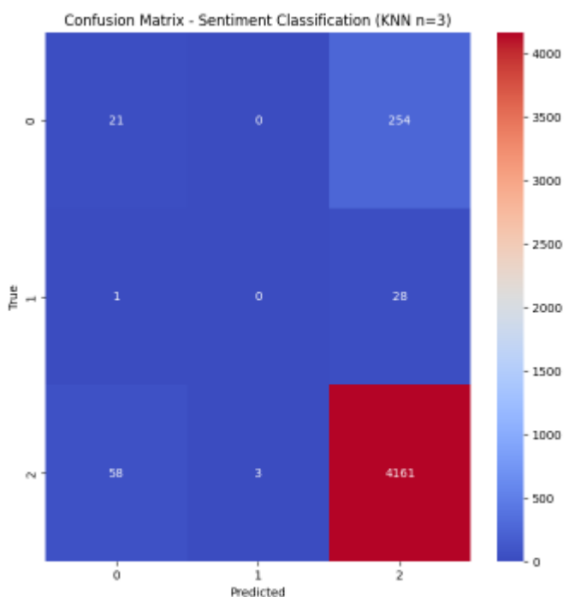
Model KNN dengan ROS ditemukan lebih unggul dibandingkan dengan SMOTE karena ROS lebih sederhana dan cepat. Metode ROS sangat sederhana karena hanya menduplikasikan sampel minoritas yang ada, membuat prosesnya cepat dan mudah untuk diimplementasikan. Sebaliknya, SMOTE, yang menciptakan sampel sintesis baru dengan interpolasi, bisa lebih lambat dan kompleks.

Secara keseluruhan, penggunaan metode oversampling dalam klasifikasi sentimen dengan KNN terbukti cukup efektif. Dengan menyeimbangkan jumlah sampel antara kelas minoritas dan mayoritas, model dapat lebih akurat dalam melakukan prediksi.

Tabel 8. Hasil Evaluasi KNN (n=3)

Model	Precision	Recall	F1-Score
Negative	0.26	0.08	0.12
Neutral	0.00	0.00	0.00
Positive	0.94	0.99	0.96
accuracy			0.92
macro avg	0.40	0.35	0.36
weighted avg	0.89	0.92	0.90

Tabel 8. menunjukkan hasil evaluasi penggunaan model KNN dengan n berjumlah 3, menghasilkan *accuracy* 0.92. Selain itu, hasil evaluasi model KNN (n=3) tanpa SMOTE dan ROS berbentuk *confusion matrix* juga ditunjukkan di Gambar 9.



Gambar 9. Confusion Matrix KNN (n = 3)

Tabel 9 menyajikan hasil evaluasi model KNN dengan SMOTE (n=3), di mana terjadi peningkatan *recall* pada kelas *Negative*, namun *precision* dan *F1-Score* secara keseluruhan menurun.

Tabel 9. Hasil Evaluasi KNN dengan SMOTE (n = 3)

Model	Precision	Recall	F1-Score
Negative	0.07	0.84	0.13
Neutral	0.01	0.34	0.02
Positive	0.98	0.06	0.11
accuracy			0.11
macro avg	0.35	0.42	0.09
weighted avg	0.92	0.11	0.11

Tabel 10 menampilkan hasil evaluasi KNN dengan *Random Oversampling* (ROS) (n=3), yang menunjukkan peningkatan pada *precision* dan *recall* untuk kelas *Positive*, namun tetap rendah pada kelas lain.

Tabel 10. Hasil Evaluasi KNN dengan ROS (n = 3)

Model	Precision	Recall	F1-Score
Negative	0.15	0.25	0.19
Neutral	0.02	0.07	0.04
Positive	0.95	0.90	0.92
accuracy			0.85
macro avg	0.37	0.49	0.38
weighted avg	0.89	0.85	0.87

3.5. Hasil Evaluasi Model

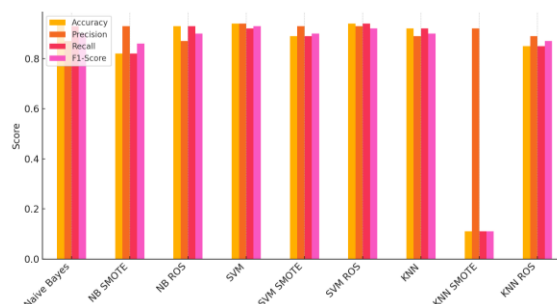
Evaluasi model menggunakan beberapa metrik seperti *accuracy*, *precision*, *recall*, dan *F1-Score*. Tabel 11 menunjukkan hasil evaluasi dari setiap model machine learning yang dilakukan. Model Naive Bayes menghasilkan nilai *accuracy* 0.933, *precision* 0.87, *recall* 0.93, dan *F1-score* sebesar 0.93.

Tabel 11. Hasil Evaluasi

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.933	0.87	0.93	0.90
NB SMOTE	0.82	0.93	0.82	0.86
NB ROS	0.93	0.87	0.93	0.90

SVM	0.94	0.94	0.92	0.93
SVM	0.89	0.93	0.89	0.90
SMOTE				
SVM ROS	0.94	0.93	0.94	0.92
KNN	0.92	0.89	0.92	0.90
KNN	0.11	0.92	0.11	0.11
SMOTE				
KNN ROS	0.85	0.89	0.85	0.87

Visualisasi hasil perbandingan berbagai macam metode dapat dilihat pada gambar 10 berikut.



Gambar 10. Grafik Perbandingan Hasil Evaluasi berbagai Python

Secara keseluruhan, model SVM memberikan nilai evaluasi tertinggi di antara semua model, ditandai dengan nilai akurasi dan *presicion* sebesar 0.94, *recall* sebesar 0.92, dan *F1-score* sebesar 0.94. Meskipun demikian, Model SVM dengan ROS memberikan hasil yang lebih seimbang, dengan performa yang mendekati SVM. Model ini mencapai akurasi sebesar 0.94 Selain itu, model ini juga memiliki *precision* sebesar 0.93 dan *recall* sebesar 0.94, yang menandakan keseimbangan yang baik antara kemampuan untuk mengidentifikasi kelas positif, netral, dan negatif. Nilai *F1-score* sebesar 0.92 cukup untuk menegaskan kinerja dan keandalan keseluruhan model ini.

4. DISKUSI

Perbandingan algoritma naive bayes, SVM, dan KNN dilakukan untuk mengklasifikasi sentimen yang ada pada ulasan produk. Namun, ketidakseimbangan data menyebabkan nilai *recall* yang sangat tinggi pada beberapa kelas tertentu terutama kelas mayoritas ("Positif"). Sehingga penggunaan metode *sampling* seperti SMOTE dan ROS menjadi upaya untuk mengatasi hal tersebut. Pada penggunaannya, teknik SMOTE dan ROS dapat mempengaruhi hasil klasifikasi.

Klasifikasi algoritma SVM dengan penerapan SMOTE meningkatkan *recall* untuk kelas "*Negative*" dari 0.12 menjadi 0.66, yang menunjukkan peningkatan sensitivitas model dalam mendeteksi ulasan negatif. Sebaliknya, ROS meningkatkan jumlah sampel kelas minoritas dengan menduplikasi data asli, yang meningkatkan *recall* untuk kelas "*Neutral*" dari 0.00 menjadi 0.21, namun tidak signifikan SMOTE. Perbedaan kinerja ini dapat dijelaskan oleh mekanisme dasar keduanya, di mana SMOTE menghasilkan variasi data yang lebih besar melalui interpolasi, sementara ROS hanya

menduplikasi data yang ada, berpotensi menyebabkan *overfitting*.

Pada penggunaan metode Naive Bayes, hanya teknik SMOTE yang menghasilkan perubahan signifikan, meningkatkan nilai *precision* dari 0.93 menjadi 0.98, namun menurunkan nilai evaluasi parameter lain. Sedangkan, teknik *random oversampling* tidak menghasilkan perubahan yang signifikan terhadap evaluasi model. Menunjukkan bahwa kedua metode sampling kurang tepat untuk diaplikasikan pada algoritma Naive Bayes.

Sedangkan untuk metode KNN, metode SMOTE merupakan metode dengan performa terburuk menurunkan akurasi sebanyak 0.81 menjadi 0.11 dari nilai akurasi sebelumnya 0.92. Sedangkan, untuk metode ROS tidak terdapat perubahan yang signifikan terhadap hasil evaluasi model.

Penelitian ini menggunakan pendekatan berbeda dengan penelitian oleh Agarap [1] yang menggunakan Bidirectional Recurrent Neural Network. Alih-alih menggunakan *deep learning*, penelitian ini mencoba pendekatan *machine learning* yang lebih sederhana seperti Naive Bayes, SVM, dan KNN, dengan penggunaan SMOTE dan ROS untuk mengakomodasi ketidakseimbangan data.

Kemudian, Penelitian ini juga memiliki perbedaan dengan penelitian yang dilakukan oleh A. Noor dan M. Islam [3] dimana masih terdapat ruang pengembangan metode pemrosesan awal *dataset* dan metode yang digunakan. Penelitian ini menggunakan TF-IDF pada proses *text processing*. Selain itu, untuk mengatasi ketidakseimbangan data digunakan dua teknik sampling yaitu SMOTE dan ROS yang tidak digunakan pada penelitian oleh A. Noor dan M. Islam [3]. Hal ini meningkatkan keseluruhan hasil akurasi metode yang digunakan pada penelitian ini menghasilkan nilai klasifikasi Naive Bayes yang meningkat sebesar 0.17 dari penelitian sebelumnya yang mendapat nilai 0.76 menjadi 0.93 pada penelitian ini. Selanjutnya, penelitian ini memberikan hasil metode yang lebih baik dengan akurasi 0.94 pada metode SVM dan 0.89 pada metode SVM SMOTE yang tidak digunakan dalam penelitian sebelumnya.

5. KESIMPULAN

Berdasarkan penelitian yang dilakukan menggunakan tiga metode berbeda (Naive Bayes, SVM, dan KNN), disimpulkan bahwa model algoritma SVM dengan ROS menunjukkan performa terbaik dibandingkan model lainnya. Hasil yang didapatkan setelah menambahkan metode ROS untuk menyeimbangkan data yang tidak seimbang adalah 0.94 untuk akurasinya, 0.93 untuk nilai *precision*, 0.94 untuk nilai *recall*, dan 0.92 untuk nilai *F1-score*. Terdapat peningkatan hasil *recall* dibandingkan dengan menggunakan metode SVM tanpa ROS. Ada beberapa saran untuk digunakan pada penelitian berikutnya yaitu, dapat menggunakan metode *resampling* data lainnya selain ROS dan SMOTE.

Lalu dapat dilakukan metode klasifikasi lainnya selain Naive Bayes, SVM, dan KNN.

DAFTAR PUSTAKA

- [1] A. F. Agarap, "Statistical Analysis on E-Commerce Reviews, with Sentiment Classification using Bidirectional Recurrent Neural Network (RNN)," Jun. 16, 2020, *arXiv*: arXiv:1805.03687. Accessed: Jun. 02, 2024. [Online]. Available: <http://arxiv.org/abs/1805.03687>
- [2] S. Sharma and S. Kumar, "Insights into the Impact of Online Product Reviews on Consumer Purchasing Decisions: A Survey-based Analysis of Brands' Response Strategies," *Sch. Int. J. Manag. Dev. ISSN 2394-3378*, vol. 10, no. 1, p. 1, Oct. 2023, doi: 10.19085/sijmd100101.
- [3] A. Noor and M. Islam, "Sentiment Analysis for Women's E-commerce Reviews using Machine Learning Algorithms," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2019, pp. 1–6. doi: 10.1109/ICCCNT45670.2019.8944436.
- [4] "A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews," *Comput. Sci. Rev.*, vol. 41, p. 100413, Aug. 2021, doi: 10.1016/j.cosrev.2021.100413.
- [5] V. Umarani, A. Julian, and J. Deepa, "Sentiment Analysis using various Machine Learning and Deep Learning Techniques," *J. Niger. Soc. Phys. Sci.*, pp. 385–394, Nov. 2021, doi: 10.46481/jnsps.2021.308.
- [6] N. M. Alharbi, N. S. Alghamdi, E. H. Alkhamash, and J. F. Al Amri, "Evaluation of Sentiment Analysis via Word Embedding and RNN Variants for Amazon Online Reviews," *Math. Probl. Eng.*, vol. 2021, pp. 1–10, May 2021, doi: 10.1155/2021/5536560.
- [7] B. Agarwal, V. Sharma, P. Harjule, V. Tiwari, and A. Sharma, "Chapter 7 - The COVID-19 outbreak: social media sentiment analysis of public reactions with a multidimensional perspective," in *Cyber-Physical Systems*, R. C. Poonia, B. Agarwal, S. Kumar, M. S. Khan, G. Marques, and J. Nayak, Eds., Academic Press, 2022, pp. 117–138. doi: 10.1016/B978-0-12-824557-6.00013-3.
- [8] "Sentiment analysis methods, applications, and challenges: A systematic literature review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 4, p. 102048, Apr. 2024, doi: 10.1016/j.jksuci.2024.102048.
- [9] J. M. Johnson and T. M. Khoshgoftaar,

- “Survey on deep learning with class imbalance,” *J. Big Data*, vol. 6, no. 1, p. 27, Mar. 2019, doi: 10.1186/s40537-019-0192-5.
- [10] “Women’s E-Commerce Clothing Reviews.” Accessed: Jun. 02, 2024. [Online]. Available: <https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews>
- [11] Md. S. Islam *et al.*, “Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach,” *Artif. Intell. Rev.*, vol. 57, no. 3, p. 62, Mar. 2024, doi: 10.1007/s10462-023-10651-9.
- [12] A. Rai and S. Borah, “Study of Various Methods for Tokenization,” in *Applications of Internet of Things*, J. K. Mandal, S. Mukhopadhyay, and A. Roy, Eds., Singapore: Springer, 2021, pp. 193–200. doi: 10.1007/978-981-15-6198-6_18.
- [13] S.-W. Kim and J.-M. Gil, “Research paper classification systems based on TF-IDF and LDA schemes,” *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, p. 30, Aug. 2019, doi: 10.1186/s13673-019-0192-7.
- [14] “An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets,” *Appl. Soft Comput.*, vol. 83, p. 105662, Oct. 2019, doi: 10.1016/j.asoc.2019.105662.
- [15] D. Elreedy and A. F. Atiya, “A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance,” *Inf. Sci.*, vol. 505, pp. 32–64, Dec. 2019, doi: 10.1016/j.ins.2019.07.070.
- [16] D. Elreedy, A. F. Atiya, and F. Kamalov, “A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning,” *Mach. Learn.*, vol. 113, no. 7, pp. 4903–4923, Jul. 2024, doi: 10.1007/s10994-022-06296-4.
- [17] M. Şimşek and A. S. Daş, “The Effect of Handling Imbalanced Datasets Methods on Prediction of Entrepreneurial Competency in University Students,” *Turk. J. Forecast.*, vol. 06, no. 2, pp. 53–60, Dec. 2022, doi: 10.34110/forecasting.1185545.
- [18] T. Hasanin, T. M. Khoshgoftaar, J. L. Leevy, and R. A. Bauder, “Severely imbalanced Big Data challenges: investigating data sampling approaches,” *J. Big Data*, vol. 6, no. 1, p. 107, Nov. 2019, doi: 10.1186/s40537-019-0274-4.
- [19] Y. A. Sir and A. H. H. Soepranoto, “Pendekatan Resampling Data Untuk Menangani Masalah Ketidakseimbangan Kelas,” *J-Icon J. Komput. Dan Inform.*, vol. 10, no. 1, Art. no. 1, Mar. 2022, doi: 10.35508/jicon.v10i1.6554.
- [20] K. Taunk, S. De, S. Verma, and A. Swetapadma, “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification,” in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India: IEEE, May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.
- [21] M. T. Khan, M. Durrani, A. Ali, I. Inayat, S. Khalid, and K. H. Khan, “Sentiment analysis and the complex natural language,” *Complex Adapt. Syst. Model.*, vol. 4, no. 1, p. 2, Dec. 2016, doi: 10.1186/s40294-016-0016-9.
- [22] M. Bansal, A. Goyal, and A. Choudhary, “A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning,” *Decis. Anal. J.*, vol. 3, p. 100071, Jun. 2022, doi: 10.1016/j.dajour.2022.100071..