Available online at https://ejournal.almaata.ac.id/index.php/IJUBI
Indonesian Journal of Business Intelligence
Volume 7 | Issue 2 | December (2024)
ISSN 2621-3915 (PRINT), ISSN 2621-3923 (ONLINE), Published by **Alma Ata University Press**

IJUBI
Indonesian Journal
---- of ----
Business Intelligence

IMPLEMENTASI SMOTE DAN ALGORITMA MACHINE LEARNING UNTUK MENINGKATKAN AKURASI REKOMENDASI HOTEL

Candra Agustina[1*], Eka Rahmawati[2], Denny Irawan[3], Vriska Wahyu Tristanti[4]
[1,4]Sistem Informasi Akuntansi Kampus Kota Surakarta, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika,
[2,3]Sistem Informasi Kampus Kota Surakarta, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika.
*candra.caa@bsi.ac.id
Jl. Kramat Raya No.98 RT.2/RW.9 Kwitang Kec. Senen, Kota Jakarta Pusat, Jakarta, Indonesia

Article history: *Received: 26 December 2024; Revised: 29 December 2024; Accepted: 31 December 2024*

Abstract

Tourism plays a significant role in the global economy, with destinations like Borobudur Temple attracting a diverse range of visitors. To enhance the visitor experience, accurate hotel recommendations are essential. However, imbalanced data, such as disproportionately positive reviews, often hampers the performance of machine learning models used for recommendations. This study aims to address this issue by applying the Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset and improve the accuracy of hotel recommendations. Various machine learning algorithms, including Random Forest, Support Vector Machines, and Neural Networks, were employed and evaluated. The results showed that the application of SMOTE significantly enhanced the performance of all models, with Random Forest yielding the best results. The study demonstrates that SMOTE, in combination with machine learning techniques, provides a robust solution to class imbalance in hotel recommendation systems, leading to more reliable and relevant recommendations for tourists. These findings have significant implications for hotel management and the broader tourism sector.

Keywords: *SMOTE, hotel recommendation, machine learning algorithms.*

Abstrak

Pariwisata memiliki peran penting dalam perekonomian global, dengan destinasi seperti Candi Borobudur menarik berbagai jenis pengunjung. Untuk meningkatkan pengalaman wisatawan, rekomendasi hotel yang akurat menjadi sangat penting. Namun, data yang tidak seimbang, seperti ulasan positif yang terlalu dominan, sering kali mengurangi kinerja model machine learning yang digunakan untuk rekomendasi. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan menerapkan Synthetic Minority Over-sampling Technique (SMOTE) guna menyeimbangkan dataset dan meningkatkan akurasi rekomendasi hotel. Beragam algoritma machine learning, termasuk Random Forest, Support Vector Machines, dan Neural Networks, diterapkan dan dievaluasi. Hasil penelitian menunjukkan bahwa penerapan SMOTE secara signifikan meningkatkan kinerja semua model, dengan Random Forest memberikan hasil terbaik. Studi ini menunjukkan bahwa SMOTE, dalam kombinasi dengan teknik machine learning memberikan solusi yang kuat terhadap ketidakseimbangan kelas pada sistem rekomendasi hotel, sehingga menghasilkan rekomendasi yang lebih andal dan relevan bagi wisatawan. Temuan ini memiliki implikasi penting bagi manajemen hotel dan sektor pariwisata secara keseluruhan.

Kata Kunci: *SMOTE, rekomendasi hotel, algoritma machine learning.*

Pendahuluan

Pariwisata telah berkembang menjadi salah satu sektor dengan pertumbuhan tercepat secara global, dengan Candi Borobudur menonjol sebagai destinasi unggulan di Indonesia. Sebagai Situs Warisan Dunia UNESCO, destinasi ini menarik beragam jenis wisatawan, sehingga menciptakan kebutuhan mendesak akan pilihan akomodasi yang sesuai dengan preferensi wisatawan yang bervariasi [1]. Rekomendasi hotel yang akurat sangat penting untuk meningkatkan pengalaman wisatawan; namun, rekomendasi ini sering menghadapi tantangan, terutama akibat dataset ulasan pengguna yang tidak seimbang [2][3]. Dalam banyak kasus, data ulasan hotel yang tersedia tidak terdistribusi secara merata, dengan jumlah ulasan positif yang jauh lebih banyak dibandingkan dengan ulasan negatif. Ketidakseimbangan ini dapat menghambat model machine learning dalam mengenali preferensi dan pola pengguna secara efektif. Untuk mengatasi masalah ini, pendekatan inovatif diperlukan, salah satunya adalah penerapan Synthetic Minority Over-sampling Technique (SMOTE) [4][5]. SMOTE meningkatkan kualitas dataset dengan menghasilkan contoh sintetik dari kelas minoritas, sehingga memberikan pandangan yang lebih komprehensif terhadap preferensi pengguna [6][7]. Penelitian ini bertujuan untuk menerapkan SMOTE bersama dengan berbagai algoritma machine learning guna meningkatkan akurasi rekomendasi hotel di sekitar kawasan Candi Borobudur. Algoritma seperti Random Forest, Support Vector Machines, dan Neural Networks akan dieksplorasi dalam penelitian ini. Dengan membandingkan kinerja berbagai algoritma tersebut, penelitian ini bertujuan untuk mengidentifikasi metode yang paling efektif dalam memberikan rekomendasi yang relevan dan memuaskan kepada pengguna. Metodologi yang digunakan dalam penelitian ini mencakup pengumpulan data ulasan hotel dari sumber-sumber terpercaya, penerapan teknik SMOTE untuk menyeimbangkan dataset, serta pelatihan model machine learning. Proses ini akan mencakup evaluasi akurasi model melalui metrik yang relevan, sehingga memberikan wawasan yang komprehensif mengenai efektivitas pendekatan yang diusulkan. Hasil dari penelitian ini diharapkan tidak hanya meningkatkan akurasi rekomendasi hotel, tetapi juga memberikan kontribusi signifikan bagi para manajer hotel dan sektor pariwisata secara lebih luas. Dengan memanfaatkan teknologi dan analitik data, penelitian ini bertujuan untuk meningkatkan pengalaman wisatawan di kawasan Candi Borobudur, sekaligus memperkuat posisi Indonesia sebagai salah satu destinasi wisata terkemuka di dunia.

Pembahasan

SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) telah digunakan secara efektif dalam berbagai aplikasi, termasuk pengenalan pola dan klasifikasi, dengan hasil yang menjanjikan [8]. Joloudari et al. menunjukkan bahwa penerapan SMOTE secara signifikan dapat meningkatkan kinerja model pada dataset yang tidak seimbang [9]. Studi lanjutan mengindikasikan bahwa mengombinasikan SMOTE dengan algoritma machine learning dapat menghasilkan akurasi yang lebih tinggi dalam sistem rekomendasi, sehingga menonjolkan utilitasnya dalam domain ini [10][11]. Ketidakseimbangan data merupakan tantangan umum dalam pengembangan model machine learning, terutama pada analisis sentimen ulasan. Sha et al. menyoroti bahwa dataset yang tidak seimbang dapat menyebabkan model lebih condong pada kelas mayoritas, yang mengurangi akurasi prediksi untuk kelas minoritas [12]. Untuk mengatasi masalah ini, teknik seperti SMOTE diperkenalkan, yang menghasilkan contoh sintetik dari kelas minoritas sehingga memungkinkan model untuk belajar dari dataset yang lebih beragam.

Machine Learning

Beragam algoritma machine learning telah diterapkan dalam sistem rekomendasi, termasuk Random Forest, Support Vector Machines (SVM), dan Neural Networks. Jarmulska mencatat bahwa Random Forest memiliki keunggulan dalam mengurangi overfitting dan memberikan interpretabilitas yang baik [13]. Sebaliknya, SVM efektif dalam menangani dataset berukuran besar dengan fitur berdimensi tinggi [14]. Neural Networks juga menunjukkan kemampuan luar biasa dalam menangkap pola kompleks dalam data [15], menjadikannya cocok untuk tugas rekomendasi yang rumit.

Algoritma Random Forest

Algoritma Random Forest adalah metode pembelajaran ensemble yang digunakan terutama untuk tugas klasifikasi dan regresi [16]. Algoritma ini bekerja dengan membangun sejumlah besar pohon keputusan selama proses pelatihan dan menghasilkan kelas yang merupakan modus dari kelas (klasifikasi) atau rata-rata prediksi (regresi) dari setiap pohon individu. Setiap pohon dibangun menggunakan subset fitur dan sampel acak dari dataset pelatihan, yang mengurangi variansi dan mencegah overfitting, masalah umum pada pohon keputusan individu. Prinsip inti dari Random Forest adalah "bagging" atau Bootstrap Aggregating, di mana beberapa pohon keputusan dilatih pada subset data yang di-bootstrap, dan prediksi akhir diperoleh dengan rata-rata atau pemungutan suara dari pohon-pohon tersebut [17]. Keanekaragaman pohon dipromosikan dengan memilih subset fitur secara acak di setiap split, memastikan bahwa model menangkap berbagai pola dalam data. Hal ini membuat Random Forest tahan terhadap noise dan mengurangi kemungkinan overfitting, sehingga meningkatkan generalisasi model.

Regresi Logistik

Regresi Logistik adalah model statistik yang umum digunakan untuk tugas klasifikasi biner [18]. Berbeda dengan regresi linear yang memprediksi hasil kontinu, regresi logistik digunakan untuk memprediksi probabilitas terjadinya suatu peristiwa biner (misalnya, sukses/gagal, 0/1, ya/tidak). Model logistik didasarkan pada fungsi logistik (sigmoid), yang mengubah kombinasi linear dari fitur input menjadi probabilitas antara 0 dan 1. Model ini mengasumsikan hubungan linear antara fitur input X dan log-odds dari hasil Y, yang dirumuskan sebagai berikut :

$$\log\left(\frac{p}{1-p}\right) = \beta 0 + \beta 1X1 + \beta 2X2 + \cdots + \beta nXn \qquad (1)$$

Dimana p adalah probabilitas bahwa peristiwa y=1 terjadi, dan $\beta 0,\beta 1,\ldots,\beta n$ adalah koefisien yang dipelajari oleh model. Fungsi logistik digunakan untuk mengubah log-odds menjadi probabilitas denga berikut:

$$p = \frac{1}{1 + e^{-\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nX_n)}} \qquad (2)$$

Fungsi ini memastikan bahwa hasil prediksi berada dalam rentang 0 hingga 1, sehingga sesuai untuk memodelkan probabilitas biner. Dengan pendekatan ini, regresi logistik menjadi alat yang sangat berguna dalam berbagai aplikasi klasifikasi, termasuk analisis sentimen, prediksi risiko, dan sistem rekomendasi. Model regresi logistik biasanya dilatih menggunakan Maximum Likelihood Estimation (MLE), di mana parameter dioptimalkan untuk memaksimalkan kemungkinan data yang diamati. Regresi logistik banyak digunakan karena kesederhanaannya, kemampuannya untuk diinterpretasikan, dan efektivitasnya dalam masalah klasifikasi biner.

KNN (K-Nearest Neighbors)

Algoritma K-Nearest Neighbors (KNN) adalah metode pembelajaran berbasis instance yang tidak parametrik dan digunakan untuk tugas klasifikasi maupun regresi[19]. KNN bekerja dengan menemukan k titik data terdekat dari data masukan berdasarkan metrik jarak yang dipilih, biasanya jarak Euclidean, dan membuat prediksi berdasarkan kelas mayoritas untuk klasifikasi atau rata-rata nilai untuk regresi.

Klasifikasi

Dalam klasifikasi, algoritma mengklasifikasikan sebuah titik data baru dengan mencari k tetangga terdekat di ruang fitur, kemudian label kelas yang paling sering muncul di antara tetangga tersebut akan diberikan pada titik data baru.

Regresi

Untuk regresi, outputnya adalah rata-rata nilai dari tetangga terdekat. Secara matematis, untuk titik query x, jarak antara x dan setiap titik data x dalam dataset pelatihan dihitung menggunakan metrik jarak d(x,xi). Pilihan metrik jarak yang umum menggunakan Euclidean:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2} \quad (3)$$

Keterangan:

X dan y : Dua titik dalam ruang n-dimensi, dimana setiap titik memiliki koordinat x dan y.

(xi-yi)2 : Selisih kuadrat antara nilai koordinat masing-masing dimensi iii untuk kedua titik.

$\sum_{i=1}^{n}$: Menjumlahkan selisih kuadrat dari semua dimensi.

√ : Mengambil akar kuadrat dari jumlah selisih kuadrat, memberikan jarak akhir dalam satuan yang sama dengan dimensi aslinya.

Algoritma KNN

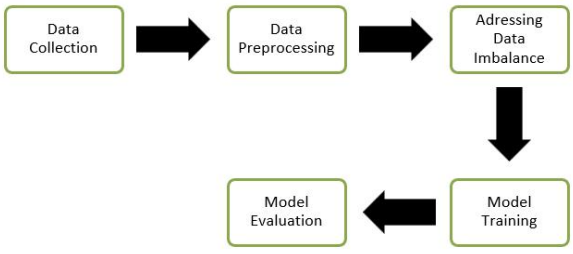
Setelah menghitung jarak antara titik query dan data dalam dataset pelatihan, algoritma memilih k data dengan jarak terkecil, dan label kelas ditentukan berdasarkan hasil pemungutan suara mayoritas di antara tetangga tersebut. KNN sederhana untuk diimplementasikan dan sering kali efektif untuk dataset kecil atau ruang fitur berdimensi rendah [20]. Namun, algoritma ini dapat menjadi mahal secara komputasi dan sensitif terhadap pilihan nilai kk dan metrik jarak, terutama pada data berdimensi tinggi. Selain itu, KNN tidak bekerja dengan baik pada dataset yang tidak seimbang atau ketika data jarang.

Rekomendasi Hotel

Rekomendasi hotel memiliki peran penting dalam meningkatkan pengalaman perjalanan, memengaruhi pilihan wisatawan, dan kepuasan secara keseluruhan. Menurut Wong et al., sistem rekomendasi yang akurat tidak hanya meningkatkan kepuasan pengguna tetapi juga meningkatkan pendapatan bagi penyedia akomodasi [21]. Sistem rekomendasi yang efektif dalam pariwisata harus mempertimbangkan berbagai faktor, termasuk preferensi individu, ulasan pengguna, dan karakteristik hotel, untuk memberikan saran yang dipersonalisasi. Penelitian sebelumnya telah mengeksplorasi penerapan teknologi rekomendasi dalam sektor pariwisata [22]. Misalnya, Fang et al. menyelidiki penggunaan deep learning untuk meningkatkan rekomendasi akomodasi [23], sementara Kim et al. menganalisis pengaruh faktor demografis terhadap preferensi hotel[24]. Namun, masih terdapat kesenjangan dalam literatur terkait penggunaan SMOTE untuk meningkatkan akurasi rekomendasi hotel, khususnya dalam konteks Borobudur.

Metode

Gambar di bawah ini menjelaskan langkah-langkah yang terlibat dalam metodologi penelitian ini, mulai dari pengumpulan data hingga evaluasi model.



Gambar 1. Langkah Penelitian

Seperti yang ditunjukkan pada Gambar 1, metodologi penelitian ini melibatkan beberapa langkah penting yang esensial dalam membangun sistem rekomendasi hotel yang andal. Langkah-langkah ini dirancang untuk memproses data, mengatasi ketidakseimbangan data, dan melatih model agar mencapai kinerja optimal. Berikut adalah penjelasan rinci dari setiap langkah yang terlibat dalam proses ini. Setiap tahap memegang peran penting dalam memastikan akurasi dan keandalan rekomendasi akhir. Proses dimulai dengan Pengumpulan Data, yang diikuti oleh beberapa fase penting lainnya sebagaimana dijelaskan di bawah ini.

1. Pengumpulan Data

Langkah pertama adalah mengumpulkan data dari berbagai sumber terpercaya yang relevan dengan rekomendasi hotel di sekitar Candi Borobudur. Hal ini mencakup scraping ulasan pengguna dan penilaian dari survei, serta mengumpulkan informasi tentang fitur hotel. Dataset harus mencakup berbagai opini pengguna untuk menangkap preferensi dan pengalaman yang beragam, sehingga menyediakan dasar analisis yang komprehensif.

2. Pra-pemrosesan Data

Setelah data terkumpul, data tersebut akan melalui tahap pra-pemrosesan untuk mempersiapkannya untuk analisis. Tahap ini mencakup pembersihan data dengan menghapus duplikasi, memperbaiki inkonsistensi, dan menangani nilai yang hilang untuk meningkatkan kualitas data. Data tekstual dari ulasan mungkin memerlukan tokenisasi, stemming, dan lemmatisasi untuk standarisasi format. Selain itu, fitur numerik akan dinormalisasi untuk memastikan konsistensi antar skala, sehingga memungkinkan pelatihan model yang lebih akurat.

3. Mengatasi Ketidakseimbangan Data (SMOTE)

Untuk menangani tantangan dataset yang tidak seimbang, terutama ketika ulasan positif lebih banyak daripada ulasan negatif, teknik Synthetic Minority Over-sampling Technique (SMOTE) diterapkan. SMOTE menghasilkan contoh sintetik untuk kelas minoritas dengan cara menginterpolasi antara contoh kelas minoritas yang sudah ada [25]. Proses ini membantu menyeimbangkan dataset, memberikan representasi yang lebih adil terhadap sentimen pengguna, dan memungkinkan model machine learning belajar lebih efektif dari semua kelas.

4. Pelatihan Model

Pada tahap ini, berbagai algoritma machine learning dilatih menggunakan dataset yang telah seimbang. Algoritma yang dipilih mencakup Random Forest, Support Vector Machines (SVM), dan Neural Networks. Dataset dibagi menjadi data pelatihan dan data pengujian untuk mendukung proses ini. Selama pelatihan, model mempelajari pola dan hubungan dalam data, serta mengoptimalkan parameter mereka untuk meningkatkan akurasi prediksi. Penyetelan hiperparameter juga dapat dilakukan untuk meningkatkan kinerja model.

5. Evaluasi Model

Setelah pelatihan, model diuji secara menyeluruh untuk menilai efektivitasnya dalam menghasilkan rekomendasi hotel yang akurat. Metrik evaluasi seperti akurasi, presisi, recall, dan F1-score digunakan untuk mengukur kinerja model [26]. Perbandingan berbagai algoritma dilakukan untuk mengidentifikasi model mana yang memberikan hasil terbaik untuk konteks spesifik rekomendasi hotel di sekitar Candi Borobudur. Evaluasi ini memberikan wawasan yang diperlukan untuk memilih model yang paling cocok untuk diterapkan dalam aplikasi dunia nyata.

Hasil

Implementasi penelitian ini dilakukan menggunakan bahasa pemrograman Python, yang sangat cocok untuk tugas analisis data dan machine learning. Kode dikembangkan dan dijalankan di Visual Studio Code (VS Code), sebuah editor kode populer yang menyediakan alat terintegrasi untuk debugging dan manajemen kode. Berbagai pustaka Python, seperti scikit-learn, pandas, dan NumPy, digunakan untuk memproses data, melatih model machine learning, serta mengevaluasi kinerjanya. Hasil detail dari eksperimen dibahas pada bagian berikut.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score
import pickle

# Load the training and testing datasets
train_data = pd.read_csv('dttrain.csv.xls')
test_data = pd.read_csv('dttest.csv.xls')

# Split the training data into features (X) and target (y)
X_train = train_data.drop('Hotel', axis=1)
y_train = train_data['Hotel']

X_test = test_data.drop('Hotel', axis=1)
y_test = test_data['Hotel']


# Initialize models
models = {
    'Random Forest': RandomForestClassifier(),
    'Logistic Regression':
LogisticRegression(max_iter=1000),
    'K-Nearest Neighbors': KNeighborsClassifier()
}

# Dictionary to store performance results
performance = {}

# Train and evaluate each model
for name, model in models.items():
    # Train the model
    model.fit(X_train, y_train)

    # Make predictions
    y_pred = model.predict(X_test)

    # Calculate performance metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred,
average='weighted')
    recall = recall_score(y_test, y_pred,
average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

# Store the performance metrics
performance[name] = {
    'Accuracy': accuracy,
    'Precision': precision,
    'Recall': recall,
    'F1 Score': f1
```

```
}

# Display the performance metrics
performance_df = pd.DataFrame(performance).T
print(performance_df)

# Find the best model based on F1 Score
best_model_name = performance_df['F1 Score'].idxmax()
best_model = models[best_model_name]

# Save the best model as a .pkl file
with open(f'{best_model_name}_model.pkl', 'wb') as file:
    pickle.dump(best_model, file)
print(f"The best model is {best_model_name} and has been
saved as {best_model_name}_model.pkl.")
```

Sebelum menerapkan SMOTE, dataset langsung digunakan untuk melatih model machine learning. Skrip Python mencakup proses pemuatan dataset, diikuti dengan langkahlangkah pra-pemrosesan data standar seperti menangani nilai yang hilang dan menormalisasi fitur numerik. Dengan menggunakan pustaka scikit-learn, tiga model—Random Forest, Logistic Regression, dan K-Nearest Neighbors (KNN)—dibangun dan dilatih pada dataset yang tidak seimbang.

Model-model ini dievaluasi menggunakan metrik kinerja standar: akurasi, presisi, recall, dan F1-score. Evaluasi awal ini bertujuan untuk memahami seberapa baik setiap model bekerja dengan data yang tidak seimbang. Hasil kinerja model sebelum penerapan SMOTE dirangkum dalam Tabel 1, yang menyoroti dampak ketidakseimbangan data terhadap akurasi dan kemampuan klasifikasi model.

Table 1. Kinerja Algoritma tanpa SMOTE Algoritma

	Accuracy	*Precision*	*Recall*	*F1 Score*
Random Forest	0.85	0.82	0.83	0.82
Logistic Regression	0.80	0.78	0.79	0.78
K-Nearest Neighbors	0.75	0.73	0.74	0.73

Dari Tabel 1, terlihat jelas bahwa model Random Forest memiliki kinerja terbaik dibandingkan dua model lainnya dalam hal akurasi dan F1 score, menunjukkan bahwa model ini memberikan prediksi yang lebih andal untuk rekomendasi hotel dalam skenario dataset yang tidak seimbang. K-Nearest Neighbors (KNN) menunjukkan kinerja terendah karena sensitivitasnya terhadap ketidakseimbangan data.

Untuk mengatasi masalah ketidakseimbangan data dan meningkatkan kinerja model, SMOTE diterapkan pada dataset menggunakan pustaka scikit-learn dalam Python. SMOTE bekerja dengan menghasilkan contoh sintetik untuk kelas minoritas, sehingga secara efektif menyeimbangkan distribusi kelas dalam set pelatihan. Dalam skrip Python, SMOTE diterapkan setelah fase pra-pemrosesan data dan sebelum pelatihan model. Skrip Python yang digunakan untuk menerapkan SMOTE dan melatih model dirancang sebagai berikut:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,
```

```
precision_score, recall_score, f1_score
from imblearn.over_sampling import SMOTE
import pickle

# Load the training and testing datasets
train_data = pd.read_csv('dttrain.csv.xls')
test_data = pd.read_csv('dttest.csv.xls')

# Split the training data into features (X) and target (y)
X_train = train_data.drop('Hotel', axis=1)
y_train = train_data['Hotel']

X_test = test_data.drop('Hotel', axis=1)
y_test = test_data['Hotel']

# Apply SMOTE to balance the training data
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote =
smote.fit_resample(X_train, y_train)

# Initialize models
models = {
'Random Forest': RandomForestClassifier(),
'Logistic Regression':
LogisticRegression(max_iter=1000),
'K-Nearest Neighbors': KNeighborsClassifier()
}

# Dictionary to store performance results
performance = {}
# Train and evaluate each model
for name, model in models.items():

# Train the model
model.fit(X_train_smote, y_train_smote)

# Make predictions
y_pred = model.predict(X_test)

# Calculate performance metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred,
average='weighted')
recall = recall_score(y_test, y_pred,
average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Store the performance metrics
performance[name] = {
'Accuracy': accuracy,
'Precision': precision,
'Recall': recall,
'F1 Score': f1
}
```

```
# Display the performance metrics
performance_df = pd.DataFrame(performance).T
print(performance_df)

# Find the best model based on F1 Score
best_model_name = performance_df['F1 Score'].idxmax()
best_model = models[best_model_name]

# Save the best model as a .pkl file
with open(f'{best_model_name}_model.pkl', 'wb') as file:
pickle.dump(best_model, file)

print(f"The best model is {best_model_name} and has been
saved as {best_model_name}_model.pkl.")
```

Dataset yang telah seimbang kemudian digunakan untuk melatih ulang tiga model machine learning—Random Forest, Logistic Regression, dan K-Nearest Neighbors (KNN) untuk mengevaluasi kinerja mereka dengan representasi kelas yang setara. Seperti yang diharapkan, penerapan SMOTE menghasilkan peningkatan yang signifikan pada semua model, terutama untuk KNN, yang sangat sensitif terhadap ketidakseimbangan kelas. Perbandingan rinci kinerja model sebelum dan sesudah penerapan SMOTE disajikan dalam Tabel 2.

Table 2. Kinerja Algoritma menggunakan SMOTE Algoritma

	Accuracy	*Precision*	*Recall*	*F1 Score*
Random Forest	0.89	0.87	0.88	0.87
Logistic Regression	0.84	0.83	0.84	0.83
K-Nearest Neighbors	0.80	0.78	0.79	0.78

Penerapan SMOTE meningkatkan kinerja semua model. Random Forest tetap menjadi model dengan kinerja terbaik, dengan peningkatan yang nyata pada akurasi dan F1 score. KNN, yang sebelumnya menunjukkan kinerja terburuk sebelum penerapan SMOTE, memperoleh manfaat yang signifikan dari teknik penyeimbangan ini, dengan peningkatan substansial pada semua metrik kinerja. Hasil penelitian menunjukkan bahwa SMOTE secara efektif meningkatkan kemampuan prediksi semua model. Peningkatan paling signifikan terlihat pada model KNN. Hal ini terjadi karena KNN sangat sensitif terhadap ketidakseimbangan kelas, yang dapat memengaruhi proses pengambilan keputusan berbasis jaraknya. Dengan menyeimbangkan distribusi kelas, SMOTE memungkinkan KNN bekerja lebih adil di semua kelas.

Model Random Forest meskipun sudah cukup tangguh dalam menangani ketidakseimbangan kelas, menunjukkan peningkatan yang sederhana setelah penerapan SMOTE. Hal ini menunjukkan kemampuan model untuk menangani dataset yang kompleks, di mana kombinasi pohon keputusan membantu mengurangi overfitting sekaligus meningkatkan akurasi klasifikasi. Dalam sistem rekomendasi hotel, pemilihan model machine learning yang tepat sangat penting untuk memastikan rekomendasi berkualitas tinggi. Dalam penelitian ini, Random Forest muncul sebagai model yang paling andal baik sebelum maupun setelah penerapan SMOTE, seperti yang dibuktikan oleh tingginya F1-score, yang menyeimbangkan presisi dan recall. Kinerja yang kuat dari model ini menunjukkan efektivitasnya dalam menangani ketidakseimbangan kelas dan interaksi fitur yang kompleks. Meskipun Logistic Regression juga menunjukkan hasil yang cukup baik, kinerjanya masih lebih rendah dibandingkan dengan Random Forest. Sifat linier dari Logistic Regression mungkin membatasi kemampuannya untuk menangkap hubungan data yang lebih rumit, yang sangat penting untuk membuat rekomendasi yang akurat. Model KNN mendapatkan manfaat paling besar dari

SMOTE, menunjukkan bahwa model ini dapat menjadi opsi yang layak jika teknik prapemrosesan yang tepat diterapkan untuk menyeimbangkan dataset. Namun, tanpa SMOTE, kinerjanya sangat terpengaruh oleh ketergantungannya pada perhitungan jarak yang sensitif terhadap ketidakseimbangan kelas.

Kesimpulan dan Saran

Penerapan SMOTE secara signifikan meningkatkan kinerja semua model machine learning. Random Forest menjadi model dengan kinerja terbaik, menunjukkan nilai akurasi dan F1-score tertinggi pada dataset yang tidak seimbang maupun yang telah seimbang. Temuan ini menekankan pentingnya mengatasi ketidakseimbangan kelas dalam pengembangan sistem rekomendasi hotel, terutama saat menggunakan algoritma seperti KNN. Penelitian di masa depan dapat difokuskan pada pengintegrasian fitur tambahan dan eksperimen dengan teknik pembelajaran ensemble lainnya untuk lebih mengoptimalkan proses rekomendasi.

Referensi

[1] A. Maulina, I. Sukoco, B. Hermanto, and N. Kostini, "Tourists' Revisit Intention and Electronic Word-of-Mouth at Adaptive Reuse Building in Batavia Jakarta Heritage," Sustainability (Switzerland), vol. 15, no. 19, Oct. 2023, doi: 10.3390/su151914227.

[2] C. Martin-Duque, J. J. Fernández-Muñoz, J. M. Moguerza, and A. Ruiz-Rua, "An empirical study on the imbalance phenomenon of data from recommendation questionnaires in the tourism sector," Journal of Tourism Futures, 2023, doi: 10.1108/JTF-09-2022-0228.

[3] B. Ray, A. Garain, and R. Sarkar, "An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews," Appl Soft Comput, vol. 98, Jan. 2021, doi: 10.1016/j.asoc.2020.106935.

[4] A. A. Nababan, M. Jannah, and A. H. Nababan, "Prediction Of Hotel Booking Cancellation Using K-Nearest Neighbors (K-Nn) Algorithm And Synthetic Minority Over-Sampling Technique (SMOTE)," Jurnal Infokum, vol. 10, no. 3, Aug. 2021, [Online]. Available: http://infor.seaninstitute.org/index.php/infokum/index

[5] M. Adil, M. F. Ansari, A. Alahmadi, J. Z. Wu, and R. K. Chakrabortty, "Solving the problem of class imbalance in the prediction of hotel cancelations: A hybridized machine learning approach," Processes, vol. 9, no. 10, Oct. 2021, doi: 10.3390/pr9101713.

[6] G. Wei, W. Mu, Y. Song, and J. Dou, "An improved and random synthetic minority oversampling technique for imbalanced data," Knowl Based Syst, vol. 248, Jul. 2022, doi: 10.1016/j.knosys.2022.108839.

[7] H. Sahlaoui, E. A. A. Alaoui, S. Agoujil, and A. Nayyar, "An empirical assessment of smote variants techniques and interpretation methods in improving the accuracy and the interpretability of student performance models," Educ Inf Technol (Dordr), vol. 29, no. 5, pp. 5447–5483, Apr. 2024, doi: 10.1007/s10639-023-12007-w.

[8] M. Umer et al., "Scientific papers citation analysis using textual features and SMOTE resampling techniques," Pattern Recognit Lett, vol. 150, pp. 250–257, Oct. 2021, doi: 10.1016/j.patrec.2021.07.009.

[9] J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere, and S. Hussain, "Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks," Applied Sciences (Switzerland), vol. 13, no. 6, Mar. 2023, doi: 10.3390/app13064006.

[10] M. Z. Abedin, C. Guotai, P. Hajek, and T. Zhang, "Combining weighted SMOTE with ensemble learning for the classimbalanced prediction of small business credit risk," Complex and Intelligent Systems, vol. 9, no. 4, pp. 3559–3579, Aug. 2023, doi: 10.1007/s40747-021-00614-4.

[11] C. Azad, B. Bhushan, R. Sharma, A. Shankar, K. K. Singh, and A. Khamparia, "Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes

mellitus," in Multimedia Systems, Springer Science and Business Media Deutschland GmbH, Aug. 2022, pp. 1289–1307. doi: 10.1007/s00530- 021-00817-2.

[12] L. Sha, M. Rakovic, A. Das, D. Gasevic, and G. Chen, "Leveraging Class Balancing Techniques to Alleviate Algorithmic Bias for Predictive Tasks in Education," IEEE Transactions on Learning Technologies, vol. 15, no. 4, pp. 481–492, Aug. 2022, doi: 10.1109/TLT.2022.3196278.

[13] Barbara. Jarmulska, Random forest versus logit models : which offers better early warning of fiscal stress? [European Central Bank], 2020.

[14] B. Gaye, D. Zhang, and A. Wulamu, "Improvement of Support Vector Machine Algorithm in Big Data Background," Math Probl Eng, vol. 2021, 2021, doi: 10.1155/2021/5594899.

[15] C. Y. Lee, H. Hasegawa, and S. Gao, "Complex-Valued Neural Networks: A Comprehensive Survey," IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 8, pp. 1406–1426, Aug. 2022, doi: 10.1109/JAS.2022.105743.

[16] G. Xu, M. Liu, Z. Jiang, D. Söffker, and W. Shen, "Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemblelearning," Sensors (Switzerland), vol. 19, no.5, Mar. 2019, doi: 10.3390/s19051088.

[17] Z. Jing, "Application of Random Forestbased supervised ensemble learning method for hail nowcasting in the Midwestern United States," 2023.

[18] A. Zaidi and A. S. M. Al Luhayb, "Two Statistical Approaches to Justify the Use of the Logistic Function in Binary Logistic Regression," Math Probl Eng, vol. 2023, no.1, Jan. 2023, doi: 10.1155/2023/5525675.

[19] P. Srisuradetchai and K. Suksrikran, "Random kernel k-nearest neighbors regression," Front Big Data, vol. 7, 2024, doi: 10.3389/fdata.2024.1402384.

[20] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing Knearest neighbor algorithm: a comprehensive review and performance analysis of modifications," J Big Data, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537- 024-00973-y.

[21] E. Wong, S. M. Rasoolimanesh, and S. Pahlevan Sharif, "Using online travel agent platforms to determine factors influencing hotel guest satisfaction," Journal of Hospitality and Tourism Technology, vol. 11, no. 3, pp. 425–445, Oct. 2020, doi: 10.1108/JHTT-07 2019-0099.

[22] A. Sarkar, T. Chowdhury, R. R. Murphy, A. Gangopadhyay, and M. Rahnemoonfar, "SAM-VQA: Supervised Attention-Based Visual Question Answering Model for Post-Disaster Damage Assessment on Remote Sensing Imagery," IEEE Transactions on Geoscience and Remote Sensing, vol. 61, 2023, doi: 10.1109/TGRS.2023.3276293.

[23] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep Learning for Sequential Recommendation," ACM Trans Inf Syst, vol. 39, no. 1, Nov. 2020, doi: 10.1145/3426723.

[24] J. Kim, D. Franklin, M. Phillips, and E. Hwang, "Online Travel Agency Price Presentation: Examining the Influence of Price Dispersion on Travelers' Hotel Preference," J Travel Res, vol. 59, no. 4, pp. 704–721, Apr. 2020, doi: 10.1177/0047287519857159.

[25] 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.

[26] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, Jan. 2020, doi: 10.1186/s12864-019-6413-7.