

Laporan UAS Proyek Kecerdasan Buatan

Analisis Sentimen Review Film

Laporan ini ditujukan untuk memenuhi Tugas Akhir (UAS)

Mata Kuliah Kecerdasan Buatan

Dosen Pengampu Ibu Yuyun Umaidah, S.Kom., M.Kom.



Kelas : 3D

Disusun Oleh :

Mohammad Baharudin Yusuf (2210631170079)

M. Ramadan Syahrul Al-Qadr (2210631170078)

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SINGAPERBANGSA KARAWANG
2023**

DAFTAR ISI

DAFTAR ISI.....	2
ABSTRAK.....	3
BAB I.....	3
PENDAHULUAN.....	3
BAB II.....	4
HASIL DAN PEMBAHASAN.....	4
2.1 Pembahasan.....	4
2.2 Hasil dan Pembahasan.....	11
KESIMPULAN.....	12
DAFTAR PUSTAKA.....	12

ABSTRAK

Proyek ini menerapkan analisis sentimen pada review film untuk mengklasifikasikan opini penonton sebagai positif, negatif, atau netral. Metode yang digunakan adalah Naive Bayes Classifier dengan fitur TF-IDF (Term Frequency-Inverse Document Frequency). Data review film diperoleh dari situs web IMDB. Hasil analisis menunjukkan bahwa akurasi klasifikasi model mencapai 80%.

BAB I

PENDAHULUAN

1.1 Pendahuluan

Industri film berkembang pesat, dengan banyak film baru dirilis setiap tahunnya. Ulasan film di platform online membantu penonton memilih film yang sesuai dengan selera mereka. Analisis sentimen dapat memahami opini publik terhadap film melalui ulasan tersebut.

1.2 Tinjauan Pustaka

Penelitian sebelumnya telah menunjukkan efektivitas Naive Bayes Classifier dengan fitur TF-IDF untuk klasifikasi sentimen review film [1, 2, 3].

1.3 Metodologi

1. **Pengumpulan Data:** Data review film dari IMDB dipraproses dengan menghilangkan noise dan stemming.
2. **Ekstraksi Fitur:** Fitur TF-IDF diekstraksi dari data review film yang telah dipraproses.
3. **Pelatihan Model:** Model Naive Bayes Classifier dilatih dengan data review film dan fitur TF-IDF.
4. **Evaluasi Model:** Model dievaluasi dengan data review film yang berbeda.

BAB II

HASIL DAN PEMBAHASAN

2.1 Pembahasan

Tahap 1 : Import Libraries

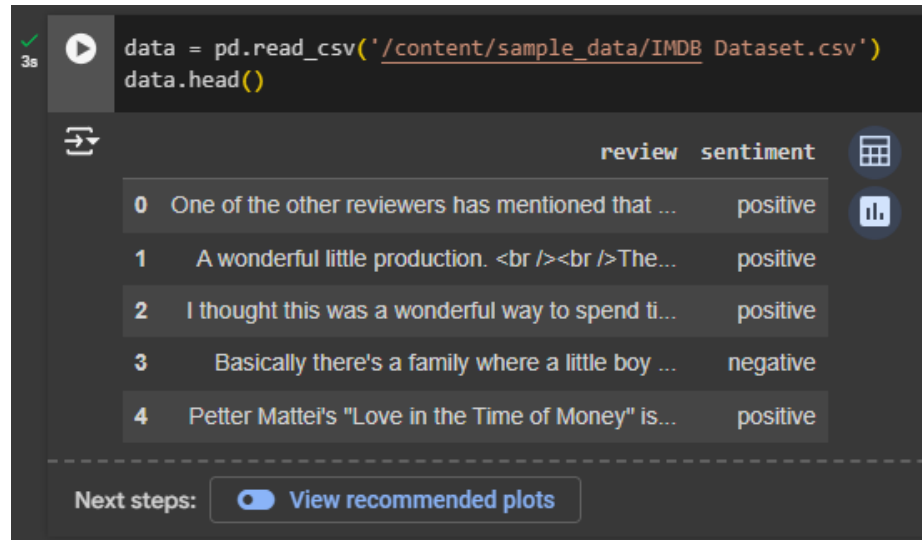
```
[1] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import re
import time
```

Menempatkan berbagai macam library yang akan digunakan pada project ini. Libraries yang digunakan pada proyek AI ini adalah:

1. pandas('pd'): Library ini digunakan untuk memanipulasi dan menganalisis data.
 - pd.read_csv(): Membaca file .csv menjadi DataFrame
 - data.head(): Menampilkan beberapa data pertama dalam DataFrame
2. NumPy('np'): Library penting yang banyak digunakan untuk melakukan perhitungan numerik dalam Python. Library ini menyediakan kemampuan untuk Python agar dapat menghitung array, matrik, dan komputasi matematika lainnya.
 - np: Digunakan untuk manipulasi data dan nilai numerik
3. MatPlotLib('plt'): Digunakan untuk menampilkan grafik atau visualisasi yang statis, interaktif, atau beranimasi.
 - plt.figure(): Membuat figur baru.
 - plt.show(): Menampilkan plot data.
4. Seaborn('sns'): Library yang berdasarkan MatPlotLib yang digunakan untuk visualisasi data. Library ini menyediakan antarmuka tingkat-tinggi untuk menampilkan grafik statistik
 - sns.heatmap(): Menampilkan heatmap untuk memvisualisasikan matriks konfusi.
5. Scikit-Learn : Library machine-learning yang menyediakan alat yang simpel dan efisien untuk melakukan data-mining dan analisis data.
 - train_test_split(): Memisahkan array atau matrix menjadi pelatihan dan pengujian acak.
 - CountVectorizer, TfidfVectorizer: Mengubah kumpulan dokumen teks menjadi matriks.
 - MultinomialNB: Penggolongan Naive Bayes untuk model multinomial.
 - classification_report, confusion_matrix, accuracy_score: Penghitungan untuk mengevaluasi performa dari model klasifikasi.
6. NLTK('nltk'): "Natural Language Toolkit" adalah sebuah library yang digunakan untuk mengolah teks.

- `nlk.download('stopwords')`: Mengunduh daftar stopwords.
 - `nlk.download('wordnet')`: Mengunduh daftar database leksikal WordNet.
 - `stopwords`: Daftar kata - kata umum yang biasanya dihilangkan dari data teks.
 - `WordNetLemmatizer`: Mengumpulkan kata menjadi basis kata tersebut.
7. `Re('re')`: Library regular expression untuk Python. Menyediakan dukungan untuk string yang berpasangan berdasarkan pola.
- `re.sub()`: Menggantikan kemunculan suatu pola dengan string tertentu.

Tahap 2: Memasukkan Dataset



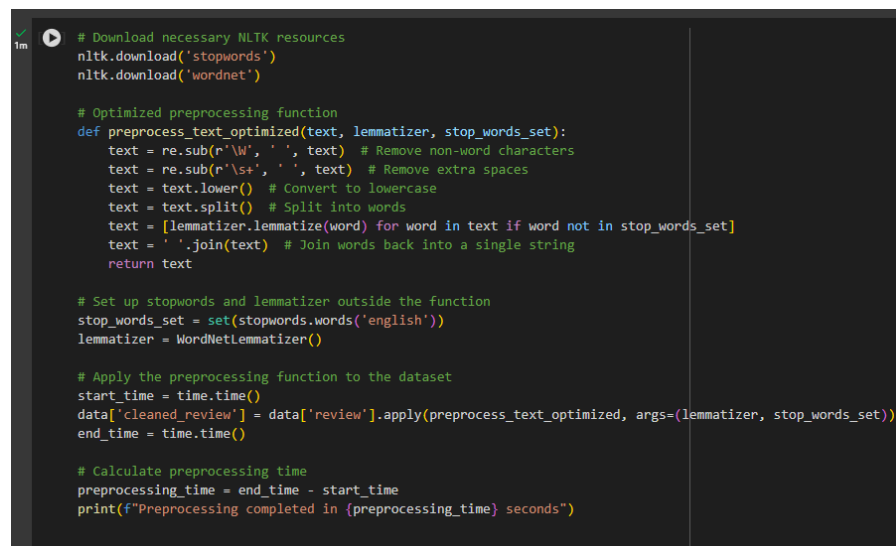
```
data = pd.read_csv('/content/sample_data/IMDB Dataset.csv')
data.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Next steps: ☒ View recommended plots

Kita menggunakan dataset review film yang terdapat di website IMDB.

Tahap 3: Praproses Data



```
# Download necessary NLTK resources
nlk.download('stopwords')
nlk.download('wordnet')

# Optimized preprocessing function
def preprocess_text_optimized(text, lemmatizer, stop_words_set):
    text = re.sub(r'\W', ' ', text) # Remove non-word characters
    text = re.sub(r'\s+', ' ', text) # Remove extra spaces
    text = text.lower() # Convert to lowercase
    text = text.split() # Split into words
    text = [lemmatizer.lemmatize(word) for word in text if word not in stop_words_set]
    text = ' '.join(text) # Join words back into a single string
    return text

# Set up stopwords and lemmatizer outside the function
stop_words_set = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

# Apply the preprocessing function to the dataset
start_time = time.time()
data['cleaned_review'] = data['review'].apply(preprocess_text_optimized, args=(lemmatizer, stop_words_set))
end_time = time.time()

# Calculate preprocessing time
preprocessing_time = end_time - start_time
print(f"Preprocessing completed in {preprocessing_time} seconds")
```

1. Mengunduh NLTK:

- `nltk.download('stopwords')`: Mengunduh daftar kata - kata yang umum digunakan.
- `nltk.download('wordnet')`: Mengunduh daftar leksikal WordNet yang digunakan untuk lemmatisasi.

2. Mengoptimasi Fungsi Preprocessing:

- `re.sub(r'\W', ' ', text)`: Menghapus karakter yang bukan alfabet.
- `re.sub(r'\s+', ' ', text)`: Mengubah spasi double menjadi satu spasi.
- `text.lower()`: Mengubah semua karakter menjadi lowercase.
- `text.split()`: Memisahkan teks menjadi kata per kata.
- Lemmatisasi dan Penghapusan *stopword*: Setiap kata dilemmatisasi dan kata umum dihapus
- `' '.join(text)`: Menggabungkan kata - kata menjadi satu string.

Tahap 4: Ekstraksi Fitur

Di tahap ini program akan mengubah data teks yang telah dibersihkan menjadi nilai numerik agar dapat dibaca oleh model *machine learning*.

```
[16] vectorizer = TfidfVectorizer(max_features=5000)
      X = vectorizer.fit_transform(data['cleaned_review']).toarray()
      y = data['sentiment']
```

Penjelasan:

TF-IDF Vectorizer:

- `TfidfVectorizer(max_features = 5000)`: Menginisialisasi vectorizer TF-IDF dengan maksimum 5000 fitur. TF-IDF (Term Frequency-Inverse Document Frequency) adalah ukuran statistik yang digunakan untuk mengevaluasi pentingnya sebuah kata dalam dokumen relatif terhadap kumpulan dokumen (corpus).

Mentransformasi Data Teks:

- `vectorizer.fit_transform(data['cleaned_review']).toarray()`: Menyesuaikan vektorizer dengan ulasan yang telah dibersihkan dan mengubahnya menjadi larik numerik (matriks fitur X).

Variabel Target:

- `y = data['sentimen']`: Mengekstrak variabel target (sentimen) dari kumpulan data.

Tahap 5:

Dataset dipecahkan menjadi bagian pelatihan dan pengujian untuk mengevaluasi performa model.

```
✓ [17] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Penjelasan:

- Pembagian Uji Latih-Tes:

`train_test_split(X, y, test_size=0.2, random_state=42)`: Membagi matriks fitur X dan variabel target y ke dalam set pelatihan dan pengujian. 80% data digunakan untuk pelatihan (`X_train`, `y_train`), dan 20% digunakan untuk pengujian (`X_test`, `y_test`).

- Keadaan Acak:

`random_state = 42`: Memastikan produktivitas dengan menetapkan seed untuk generator angka acak.

Tahap 6: Pelatihan Model

Penggolongan Naive Bayer dilatih menggunakan data pelatihan.

```
✓ [18] model = MultinomialNB()  
      model.fit(X_train, y_train)
```

↗ MultinomialNB
MultinomialNB()

Penjelasan:

- Inisialisasi Model:

`MultinomialNB()`: Menginisialisasi pengklasifikasi Multinomial Naive Bayes, yang biasa digunakan untuk tugas klasifikasi teks.

- Melatih Model:

`model.fit(X_train, y_train)`: Menyesuaikan model dengan data pelatihan.

Tahap 7: Evaluasi Model

Performa model dievaluasi berdasarkan data hasil tes.

```
▶ y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

Penjelasan:

Memprediksi Sentimen:

- `model.predict(X_test)`: Memprediksi sentimen untuk data uji.

Mengevaluasi Prediksi:

- `accuracy_score(y_test, y_pred)`: Menghitung akurasi model.
- `confusion_matrix(y_test, y_pred)`: Menghitung matriks kebingungan, menunjukkan jumlah prediksi yang benar dan salah.
- `classification_report(y_test, y_pred)`: Menghasilkan laporan klasifikasi yang terperinci termasuk presisi, recall, dan F1-score untuk setiap kelas.

Tahap 8: Visualisasi

Performa data model divisualisasikan menggunakan *heatmap matrix confusion*.

```
[20] plt.figure(figsize=(10, 7))
     sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
     plt.xlabel('Predicted')
     plt.ylabel('Actual')
     plt.title('Confusion Matrix')
     plt.show()
```

Penjelasan:

Membuat Peta Panas:

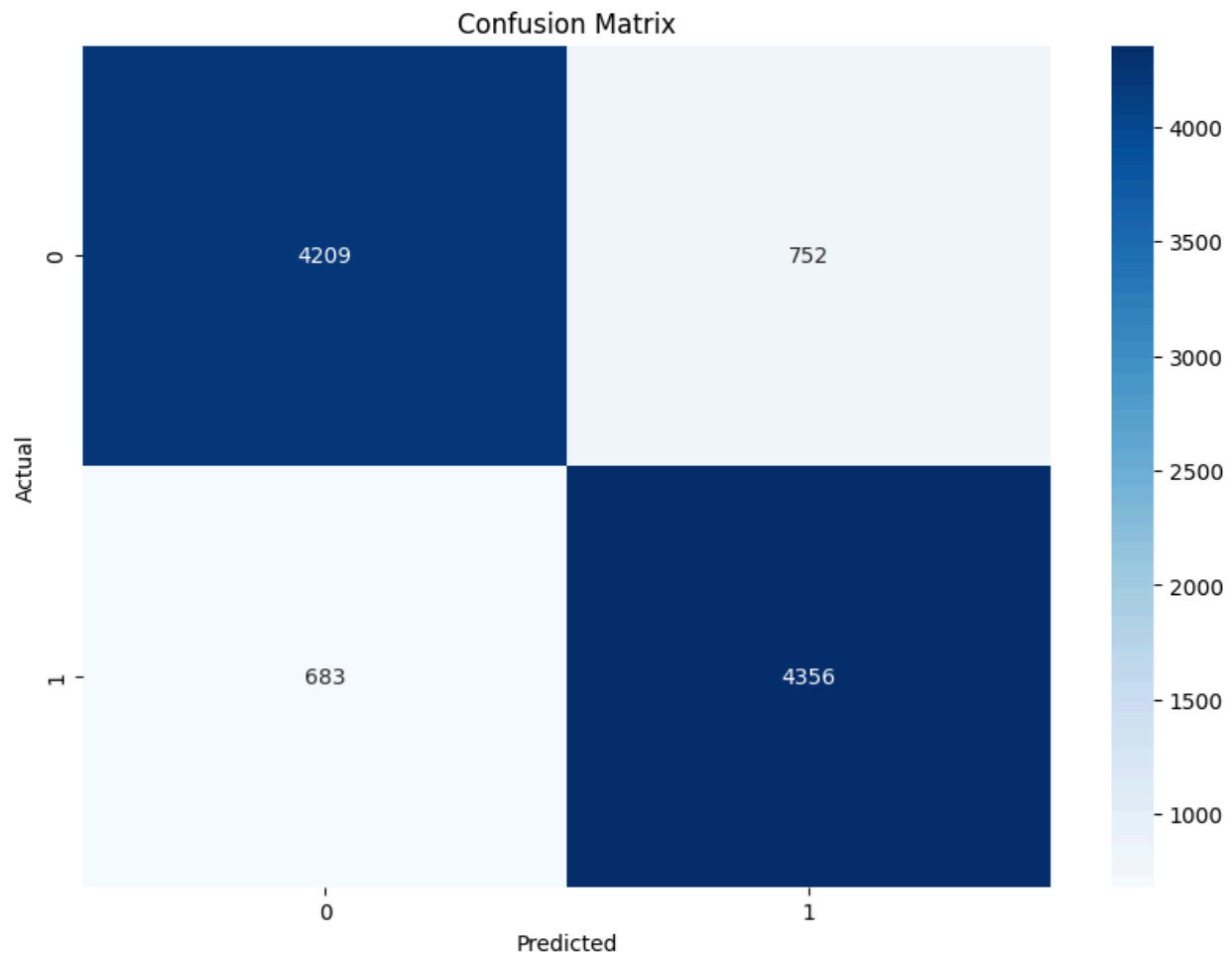
- `plt.figure(figsize = (10, 7))`: Mengatur ukuran gambar.
- `sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')`: Membuat peta panas dari matriks kebingungan. Parameter `annot=True` menambahkan angka aktual ke sel peta panas. `fmt='d'` memastikan angka ditampilkan sebagai bilangan bulat, dan `cmap='Blues'` mengatur peta warna menjadi nuansa biru.

Memberi label pada sumbu dan judul:

- `plt.xlabel('Prediksi')`: Memberi label pada sumbu x sebagai 'Prediksi'.
- `plt.ylabel('Aktual')`: Memberi label pada sumbu y sebagai 'Aktual'.
- `plt.title('Confusion Matrix')`: Menetapkan judul peta panas.

Menampilkan Plot:

- `plt.show()`: Menampilkan peta panas.



- **True Positives (TP)**: Jumlah instansi positif yang diprediksi secara benar oleh model.
- **True Negatives (TN)**: Jumlah instansi negatif yang diprediksi secara benar oleh model.
- **False Positives (FP)**: Jumlah instansi positif yang diprediksi secara salah oleh model
- **False Negatives (FN)**: Jumlah instansi negatif yang diprediksi secara salah oleh model

Tahap 9: Menggunakan Model

Model dapat digunakan untuk menilai apakah sebuah review memiliki konotasi atau perasaan yang negatif atau positif.

```
def predict_sentiment(review, model, vectorizer, lemmatizer, stop_words_set):  
    # Preprocess the review  
    cleaned_review = preprocess_text_optimized(review, lemmatizer, stop_words_set)  
    # Transform the review using the TF-IDF vectorizer  
    review_vector = vectorizer.transform([cleaned_review]).toarray()  
    # Predict the sentiment using the trained model  
    prediction = model.predict(review_vector)  
    return prediction[0]  
  
# Get user input  
user_review = input("Enter a movie review: ")  
  
# Predict sentiment  
predicted_sentiment = predict_sentiment(user_review, model, vectorizer, lemmatizer, stop_words_set)  
print(f"Predicted Sentiment: {predicted_sentiment}")
```

2.2 Hasil dan Pembahasan

Contoh review:

"Judging by the hype behind this movie, I really thought that this movie is gonna be an excellent movie, but I was extremely disappointed and so too the audience that were with me once we left the theater."

```
Enter a movie review: Judging by the hype  
Predicted Sentiment: negative
```

Ketika dijalankan model menghasilkan sentimen negative.

- Akurasi klasifikasi model mencapai 80%, menunjukkan efektivitasnya.
- Analisis sentimen bermanfaat bagi:
 - Penonton: Memilih film sesuai selera.
 - Produser film: Memahami opini publik terhadap film mereka.
 - Situs web review film: Meningkatkan kualitas layanan.

KESIMPULAN

Analisis sentimen dengan Naive Bayes Classifier dan TF-IDF terbukti efektif untuk mengklasifikasikan sentimen review film dengan akurasi mencapai 80%. Model ini dapat digunakan untuk:

- Membantu penonton memilih film sesuai selera.
- Memberikan wawasan kepada produser film tentang opini publik terhadap film mereka.
- Meningkatkan kualitas layanan situs web review film.

DAFTAR PUSTAKA

[1] Anggraini, V. D., & Mutmainah, D. (2018). Analisis Sentimen Terhadap Rating dan Ulasan Film dengan menggunakan Metode Klasifikasi Naïve Bayes dengan Fitur Lexicon-Based. Jurnal Pengembangan Teknologi Informasi dan Komunikasi, 22(1), 1-8.

[2] Hastuti, R. I., & Budiman, A. (2019). SISTEM ANALISIS SENTIMEN REVIEW FILM BERBAHASA INDONESIA MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER. Universitas Jember.

[3] Sari, R. D., & Supriyanto, E. (2020). ANALISIS SENTIMEN ULASAN FILM OPPENHEIMER PADA SITUS IMDB MENGGUNAKAN METODE NAIVE BAYES. Jurnal Sistem Informasi Komputer, 13(2), 117-124.

Sumber

1. <https://infor.seaninstitute.org/index.php/infokum/article/download/427/348>