Analisis sentimen pada dataset amazon_reviews.csv

Oleh : Dodo Zaenal Abidin

Sumber : https://www.kaggle.com/code/mehmetisik/sentiment-analysis-and-modeling-for-amazon/notebook

Sumber : https://www.feedbackwhiz.com/blog/how-to-get-effective-product-reviews-on-amazon-in-2020/

Tahapan Analisis Sentimen

1. Pengenalan dan Eksplorasi Data:

- Melihat struktur data dan memahami konten dari dataset.
- Mengidentifikasi missing values dan melihat distribusi rating.

2. Preprocessing Teks:

- Menghapus missing values dalam teks ulasan.
- Membersihkan teks ulasan (mengonversi ke huruf kecil, menghapus tanda baca dan karakter khusus, menghapus stop words).

3. Label Sentimen:

- Menentukan label sentimen berdasarkan rating (overall). Misalnya, rating 4-5 sebagai positif, 3 sebagai netral, dan 1-2 sebagai negatif.

4. Ekstraksi Fitur:

- Mengubah teks ulasan menjadi representasi numerik menggunakan teknik TF-IDF.

5. Pembagian Dataset:

- Membagi data menjadi data latih dan data uji untuk memvalidasi performa model.

6. Pelatihan Model:

- Melatih model Naive Bayes menggunakan data latih.

7. Prediksi dan Evaluasi:

- Menggunakan model untuk memprediksi sentimen pada data uji.
- Mengevaluasi performa model menggunakan metrik seperti akurasi, precision, recall, dan F1-score.

Kode Lengkap untuk Analisis Sentimen Menggunakan Naive Bayes :

Langkah 1: Import Pustaka yang Diperlukan

```
# Langkah 1: Import Pustaka yang Diperlukan
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
import string
import matplotlib.pyplot as plt
import seaborn as sns
```

Langkah 2: Baca Dataset

```
# Langkah 2: Baca Dataset
file_path = '/content/drive/MyDrive/NLP/NLP13/amazon_reviews.csv'
df = pd.read_csv(file_path)
```

Langkah 3: Eksplorasi Data

```
print("Info Data:")
```

```
Info Data:
```

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4915 entries, 0 to 4914
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   reviewerID      4915 non-null   object
 1   asin            4915 non-null   object
 2   reviewerName    4914 non-null   object
 3   helpful         4915 non-null   object
 4   reviewText      4914 non-null   object
 5   overall         4915 non-null   float64
 6   summary         4915 non-null   object
 7   unixReviewTime  4915 non-null   int64
 8   reviewTime      4915 non-null   object
 9   day_diff        4915 non-null   int64
 10  helpful_yes     4915 non-null   int64
 11  total_vote      4915 non-null   int64
dtypes: float64(1), int64(4), object(7)
memory usage: 460.9+ KB
None
```

```
print("\nDistribusi Rating:")
```

```
Distribusi Rating:
```

```
print(df['overall'].value_counts())
```

```
overall
5.0    3922
4.0     527
1.0     244
3.0     142
2.0      80
Name: count, dtype: int64
```

Langkah 4: Preprocessing Teks

```
# Menghapus missing values pada reviewText
df_clean = df.dropna(subset=['reviewText'])
```

```
# Mengonversi teks ke huruf kecil
df_clean['reviewText'] = df_clean['reviewText'].str.lower()
```

```
<ipython-input-27-45bf0a42b6b4>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
  df_clean['reviewText'] = df_clean['reviewText'].str.lower()
```

```
print(df_clean['reviewText'])
```

```
0                                                no issues.
1       purchased this for my device, it worked as adv...
2       it works as expected. i should have sprung for...
3       this think has worked out great.had a diff. br...
4       bought it with retail packaging, arrived legit...
                                ...
4910    i bought this sandisk 16gb class 10 to use wit...
4911    used this for extending the capabilities of my...
4912    great card that is very fast and reliable. it ...
4913    good amount of space for the stuff i want to d...
4914    i've heard bad things about this 64gb micro sd...
Name: reviewText, Length: 4914, dtype: object
```

```
# Menghapus tanda baca dan karakter khusus
df_clean['reviewText'] = df_clean['reviewText'].apply(
    lambda x: x.translate(str.maketrans('', '', string.punctuation))
)
```

```
<ipython-input-33-a9965ba342b2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
  df_clean['reviewText'] = df_clean['reviewText'].apply(
```

```
print(df_clean['reviewText'])
```

```
0                                                no issues
1       purchased this for my device it worked as adve...
2       it works as expected i should have sprung for ...
3       this think has worked out greathad a diff bran...
4       bought it with retail packaging arrived legit ...
                                ...
4910    i bought this sandisk 16gb class 10 to use wit...
4911    used this for extending the capabilities of my...
4912    great card that is very fast and reliable it c...
4913    good amount of space for the stuff i want to d...
4914    ive heard bad things about this 64gb micro sd ...
Name: reviewText, Length: 4914, dtype: object
```

```
# Menghapus stop words (dengan daftar stop words yang sederhana)
stop_words = set([
    'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves',
    'he', 'him', 'his', 'himself', 'she', 'her', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their',
    'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was',
    'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and',
    'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between',
    'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off',
    'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any',
    'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so',
    'than', 'too', 'very', 's', 't', can', 'will', 'just', 'don', 'should', 'now'
])
```

```
df_clean['reviewText'] = df_clean['reviewText'].apply(
    lambda x: ' '.join([word for word in x.split() if word not in stop_words])
)
```

```
<ipython-input-36-37e90884599f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
  df_clean['reviewText'] = df_clean['reviewText'].apply(
```

```
print(df_clean['reviewText'])
```

```
0                                                 issues
1       purchased device worked advertised never much ...
2       works expected sprung higher capacity think ma...
3       think worked greathad diff bran 64gb card went...
4       bought retail packaging arrived legit orange e...
                                ...
4910    bought sandisk 16gb class 10 use htc inspire 3...
4911    used extending capabilities samsung galaxy not...
4912    great card fast reliable comes optional adapte...
4913          good amount space stuff want fits gopro say
4914    ive heard bad things 64gb micro sd card crappi...
Name: reviewText, Length: 4914, dtype: object
```

Langkah 5: Label Sentimen

```
# Langkah 5: Label Sentimen
def sentiment_label(rating):
    if rating >= 4:
        return 'positive'
    elif rating == 3:
        return 'neutral'
    else:
        return 'negative'

df_clean['sentiment'] = df_clean['overall'].apply(sentiment_label)
```

```
<ipython-input-41-ba5ad4da8229>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
  df_clean['sentiment'] = df_clean['overall'].apply(sentiment_label)
```

Langkah 6: Ekstraksi Fitur dengan TF-IDF

```
tfidf_vectorizer = TfidfVectorizer(max_features=1000)
X = tfidf_vectorizer.fit_transform(df_clean['reviewText'])
```

```
# Label sentimen
y = df_clean['sentiment']
```

Langkah 7: Pembagian Dataset

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

Langkah 8: Pelatihan Model Naive Bayes

```
# Langkah 8: Pelatihan Model Naive Bayes
model = MultinomialNB()
model.fit(X_train, y_train)
```

```
MultinomialNB
MultinomialNB()
```

Langkah 9: Prediksi dan Evaluasi

```
y_pred = model.predict(X_test)
```

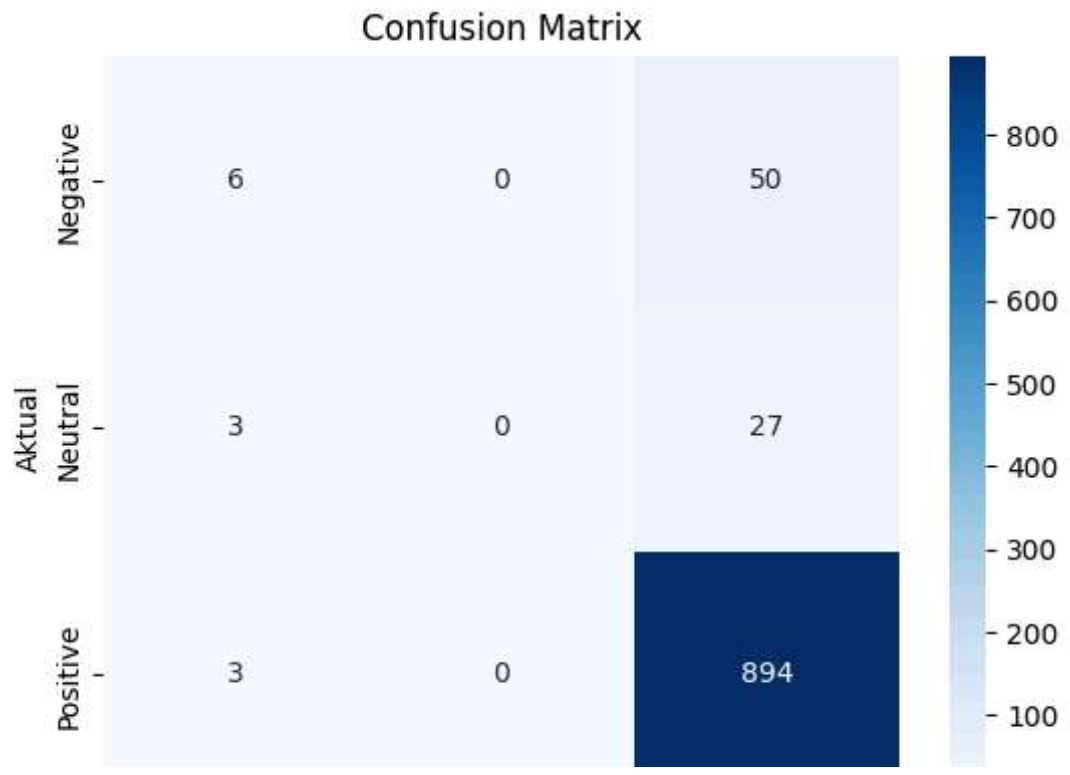
```
# Menampilkan hasil evaluasi
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

    negative       0.50      0.11      0.18        56
     neutral       0.00      0.00      0.00        30
    positive       0.92      1.00      0.96       897

    accuracy                           0.92       983
   macro avg       0.47      0.37      0.38       983
weighted avg       0.87      0.92      0.88       983

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score a
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score a
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score a
  _warn_prf(average, modifier, msg_start, len(result))
```

```
print("Confusion Matrix:")
conf_matrix = confusion_matrix(y_test, y_pred)
print(conf_matrix)
```

```
Confusion Matrix:
[[  6   0  50]
 [  3   0  27]
 [  3   0 894]]
```

```
# Visualisasi Confusion Matrix
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Neutral', 'Positive'], yticklabels=['Negative'
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix')
plt.show()
```


Confusion Matrix
Aktual
Negative
Neutral
Positive
6
0
50
3
0
27
3
0
894
800
700
600
500
400
300
200
100