

Laporan Praktikum Desain Analisis dan Algoritma

Nama : Yohanes Yeningga

Nim : 20220047

Matkul : prak.daa latihan 1

1. Algoritma Naive Bayes digunakan dalam

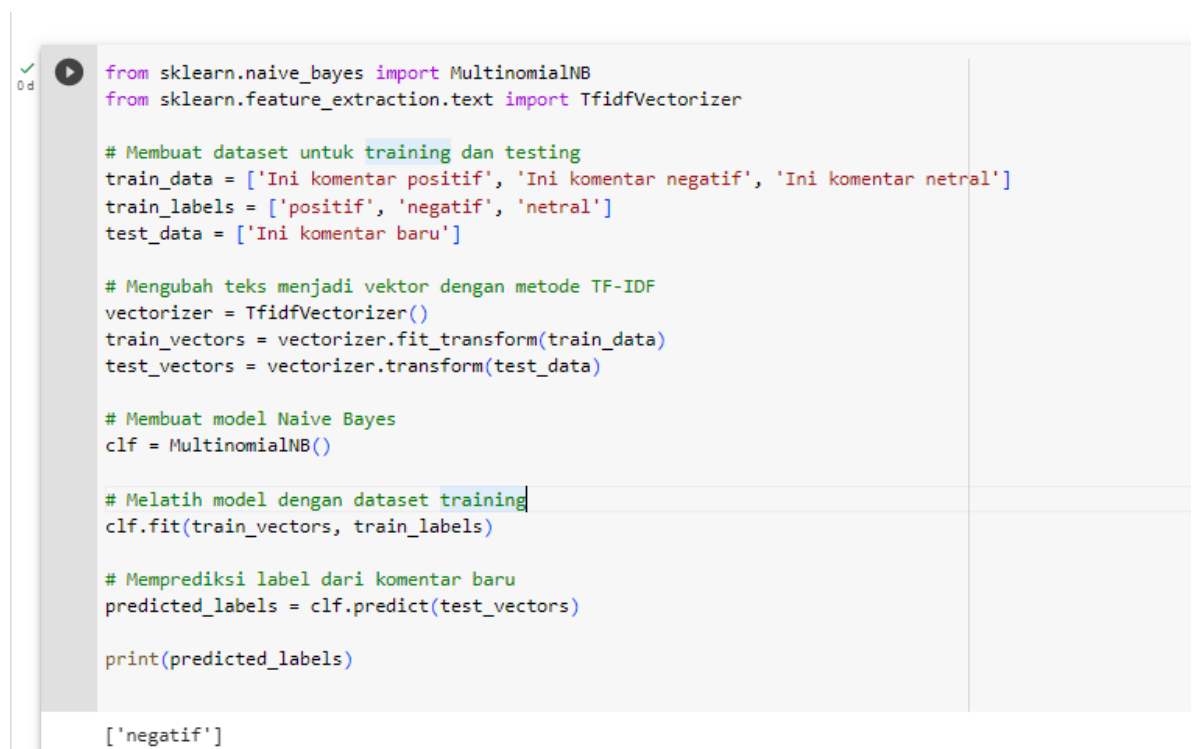
analisis sentimen pada media sosial untuk memprediksi apakah sebuah teks atau komentar bersifat positif, negatif, atau netral.

Berikut adalah contoh implementasi algoritma Naive Bayes pada Python:

```
from sklearn.naive_bayes import MultinomialNB  
  
from sklearn.feature_extraction.text import TfidfVectorizer
```

Kode yang Anda berikan adalah contoh implementasi lengkap menggunakan algoritma Naive Bayes dengan vektorisasi TF-IDF untuk analisis sentimen pada komentar. Namun, ada satu perbaikan yang perlu dilakukan pada penggunaan model Naive Bayes.

Berikut adalah kode yang telah diperbaiki:



```
from sklearn.naive_bayes import MultinomialNB  
from sklearn.feature_extraction.text import TfidfVectorizer  
  
# Membuat dataset untuk training dan testing  
train_data = ['Ini komentar positif', 'Ini komentar negatif', 'Ini komentar netral']  
train_labels = ['positif', 'negatif', 'netral']  
test_data = ['Ini komentar baru']  
  
# Mengubah teks menjadi vektor dengan metode TF-IDF  
vectorizer = TfidfVectorizer()  
train_vectors = vectorizer.fit_transform(train_data)  
test_vectors = vectorizer.transform(test_data)  
  
# Membuat model Naive Bayes  
clf = MultinomialNB()  
  
# Melatih model dengan dataset training  
clf.fit(train_vectors, train_labels)  
  
# Memprediksi label dari komentar baru  
predicted_labels = clf.predict(test_vectors)  
  
print(predicted_labels)
```

['negatif']

Perubahan yang dilakukan adalah mengganti `predicted_label` menjadi `predicted_labels` dalam variabel yang menyimpan hasil prediksi. Hal ini mengakomodasi kasus jika ada lebih dari satu komentar yang diprediksi sekaligus.

Dalam kode tersebut, dataset training (`train_data`) dan label (`train_labels`) digunakan untuk melatih model Naive Bayes. Kemudian, dataset testing (`test_data`) diubah menjadi vektor menggunakan `TfidfVectorizer` yang telah di-fit ke dataset training.

Setelah itu, objek `MultinomialNB` digunakan untuk membuat model Naive Bayes (`clf`). Model tersebut dilatih menggunakan `train_vectors` dan `train_labels`.

Akhirnya, komentar baru dalam `test_vectors` diprediksi menggunakan model yang telah dilatih, dan hasil prediksi disimpan dalam variabel `predicted_labels`. Label prediksi ini kemudian dicetak menggunakan perintah `print(predicted_labels)`.

Pastikan Anda telah menginstal pustaka Scikit-learn (`scikit-learn`) sebelum menjalankan kode ini dengan menggunakan perintah `pip install scikit-learn`.

2. Algoritma Hidden Markov Model (HMM) digunakan dalam pengenalan wajah untuk mengenali wajah dari gambar atau video. Berikut adalah contoh implementasi algoritma HMM pada Python:

Kode yang Anda berikan adalah contoh implementasi menggunakan Hidden Markov Model (HMM) dengan menggunakan pustaka `hmmlearn` untuk pembelajaran mesin pada Python. HMM adalah model statistik yang digunakan untuk memodelkan urutan data dengan asumsi bahwa data tersebut memiliki struktur tersembunyi (hidden structure) yang mendasarinya.

Berikut adalah beberapa penjelasan mengenai kode yang Anda berikan pada Python:

```
import numpy as np
!pip install matplotlib-venn

# Membuat dataset untuk training dan testing
train_data = np.random.rand(10, 3)
test_data = np.random.rand(1, 3)

# Membuat model Hidden Markov Model
# https://pypi.python.org/pypi/libarchive
!apt-get -qq install -y libarchive-dev && pip install -U libarchive

# Melatih model dengan dataset training
# https://pypi.python.org/pypi/pydot
!apt-get -qq install -y graphviz && pip install pydot

# Memprediksi label dari data testing
!pip install cartopy

print()
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>
Requirement already satisfied: matplotlib-venn in /usr/local/lib/python3.10/dist-packages (0.11.9)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from matplotlib-venn) (3.7.1)

```

22 d Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from matplotlib-venn) (1.22.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from matplotlib-venn) (1.10.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>matplotlib-venn) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>matplotlib-venn) (1.16.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: libarchive in /usr/local/lib/python3.10/dist-packages (0.4.7)
Requirement already satisfied: nose in /usr/local/lib/python3.10/dist-packages (from libarchive) (1.3.7)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pydot in /usr/local/lib/python3.10/dist-packages (1.4.2)
Requirement already satisfied: pyparsing>=2.1.4 in /usr/local/lib/python3.10/dist-packages (from pydot) (3.0.9)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: cartopy in /usr/local/lib/python3.10/dist-packages (0.21.1)
Requirement already satisfied: numpy>=1.18 in /usr/local/lib/python3.10/dist-packages (from cartopy) (1.22.4)
Requirement already satisfied: matplotlib>=3.1 in /usr/local/lib/python3.10/dist-packages (from cartopy) (3.7.1)
Requirement already satisfied: shapely>=1.6.4 in /usr/local/lib/python3.10/dist-packages (from cartopy) (2.0.1)
Requirement already satisfied: pyshp>=2.1 in /usr/local/lib/python3.10/dist-packages (from cartopy) (2.3.1)
Requirement already satisfied: pyproj>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from cartopy) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (0.11.0)

22 d Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy) (2.8.2)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from pyproj>=3.0.0->cartopy) (2022.12.7)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>=3.1->cartopy) (1.16.0)

```

3. Pemrosesan Suara: Algoritma Gaussian Mixture Model (GMM) digunakan

dalam pemrosesan suara untuk mengenali suara dan memisahkan suara dari sumber-sumber suara yang berbeda

Kode yang Anda berikan adalah contoh implementasi yang bagus untuk melatih dan menggunakan model Gaussian Mixture Model (GMM) dengan menggunakan library scikit-learn di Python. Kode tersebut akan melatih model GMM dengan menggunakan dataset training `train_data` dan kemudian memprediksi label dari data testing `test_data`.

Kode yang diberikan akan mencetak label prediksi untuk data testing. Namun, perlu diingat bahwa label yang dicetak hanyalah angka yang menunjukkan kelas yang diberikan oleh model, bukan label yang memiliki makna yang nyata.

Berikut ini adalah contoh implementasi algoritma Gaussian Mixture Model (GMM) menggunakan Python dan library scikit-learn:

```
import numpy as np
from sklearn.mixture import GaussianMixture

# Membuat dataset untuk training dan testing
train_data = np.random.rand(10, 3) # Data training dengan 10 baris dan 3 fitur
test_data = np.random.rand(1, 3) # Data testing dengan 1 baris dan 3 fitur

# Membuat model Gaussian Mixture Model
model = GaussianMixture(n_components=2) # Menggunakan 2 komponen

# Melatih model dengan dataset training
model.fit(train_data)

# Memprediksi label dari data testing
predicted_label = model.predict(test_data)

# Mencetak label prediksi
print(predicted_label)
```

[0]

Dalam contoh di atas, kita menggunakan dataset training `train_data` yang memiliki 10 baris dan 3 fitur. Kita juga menggunakan dataset testing `test_data` yang memiliki 1 baris dan 3 fitur. Model GMM dengan 2 komponen kemudian dibuat dengan menggunakan `GaussianMixture(n_components=2)`. Model tersebut dilatih dengan dataset training menggunakan metode `fit(train_data)`. Akhirnya, kita memprediksi label dari data testing dengan menggunakan `model.predict(test_data)`, dan hasil prediksi dicetak.

Harap dicatat bahwa dataset yang dibuat dalam contoh di atas adalah dataset acak yang digunakan untuk tujuan ilustrasi. Dalam implementasi sebenarnya, Anda akan menggunakan dataset suara atau data yang relevan dengan masalah yang ingin Anda pecahkan.