

Nama : Febi Rahmadia Putri

Nim : 20220021

Prak1.Algoritma Prabolistik

1. Pada contoh implementasi tersebut, langkah-langkah yang dilakukan adalah sebagai berikut:

1. Mengimpor kelas **MultinomialNB** dari modul **naive_bayes** dan kelas **TfidfVectorizer** dari modul **feature_extraction.text** dari library **scikit-learn**.
2. Membuat dataset untuk pelatihan (**train_data**) dan pengujian (**test_data**). Dataset pelatihan terdiri dari beberapa teks komentar dan label sentimen yang sesuai.
3. Membuat objek **TfidfVectorizer** yang akan mengubah teks menjadi vektor dengan metode TF-IDF. TF-IDF adalah singkatan dari "Term Frequency-Inverse Document Frequency" yang merupakan metode yang umum digunakan untuk menghitung bobot kata dalam korpus teks.
4. Mengubah data pelatihan (**train_data**) menjadi vektor menggunakan metode **fit_transform** dari objek **TfidfVectorizer**. Data pengujian (**test_data**) juga diubah menjadi vektor menggunakan metode **transform** dari objek yang sama.
5. Membuat model Naive Bayes dengan menginstansiasi objek **MultinomialNB**.
6. Melatih model Naive Bayes dengan memanggil metode **fit** pada objek model dengan menggunakan vektor data pelatihan dan label pelatihan (**train_labels**).
7. Memprediksi label sentimen dari data pengujian dengan memanggil metode **predict** pada objek model dengan menggunakan vektor data pengujian (**test_vectors**).
8. Menampilkan hasil prediksi label sentimen dengan mencetak nilai **predicted_label**.

Dalam contoh tersebut, teks "Ini komentar baru" akan diprediksi label sentimennya menggunakan model Naive Bayes yang telah dilatih. Hasil prediksi akan dicetak pada baris terakhir menggunakan pernyataan **print(predicted_label)**.

Berikut percobaan menggunakan python :

```
1d 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_label = clf.predict(test_vectors)

    print(predicted_label)
```

['negatif']

Activate \

2. Pada contoh implementasi tersebut, langkah-langkah yang dilakukan adalah sebagai berikut:

1. Mengimpor modul **numpy** dan kelas **hmm** dari library **hmmlearn**.
2. Membuat dataset untuk pelatihan (**train_data**) dan pengujian (**test_data**). Dataset pelatihan berisi sejumlah fitur numerik yang mewakili wajah, dan dataset pengujian berisi satu set fitur numerik yang akan diprediksi labelnya.
3. Membuat model Hidden Markov Model dengan menginstansiasi objek **GaussianHMM** dari kelas **hmm.GaussianHMM**. Pada contoh ini, model memiliki 2 komponen dan tipe kovariansi "full".
4. Melatih model dengan memanggil metode **fit** pada objek model dengan menggunakan data pelatihan (**train_data**).
5. Memprediksi label dari data pengujian dengan memanggil metode **predict** pada objek model dengan menggunakan data pengujian (**test_data**).
6. Menampilkan hasil prediksi label dengan mencetak nilai **predicted_label**.

Dalam contoh ini, label yang diprediksi untuk data pengujian dicetak pada baris terakhir menggunakan pernyataan **print(predicted_label)**.

3. Implementasi yang Anda berikan adalah contoh sederhana dari penggunaan algoritma Gaussian Mixture Model (GMM) dengan library sklearn di Python untuk memprediksi label dari data testing. GMM adalah model probabilitas yang digunakan untuk pemodelan distribusi data dengan asumsi bahwa data berasal dari campuran beberapa distribusi Gaussian.