Laporan Praktikum Desain Analisis dan Algoritma

Nama : Yohanes Yeningga

Nim : 20220047

Matkul : prak.daa latihan 2

1. Algoritma probabilistik adalah algoritma yang menggunakan konsep probabilitas untuk memodelkan atau memecahkan masalah. Algoritma ini berfokus pada penggunaan probabilitas untuk mengambil keputusan atau menghasilkan solusi yang optimal.
   Dalam algoritma probabilistik, variabel-variabel dalam masalah yang ingin dipecahkan dianggap sebagai variabel acak yang memiliki distribusi probabilitas. Algoritma ini menggunakan informasi probabilitas untuk memperkirakan atau menghitung kemungkinan kejadian tertentu dan memilih tindakan yang paling optimal berdasarkan informasi tersebut.

   Contoh penerapan algoritma probabilistik dalam kehidupan nyata adalah:
   1) Deteksi Spam: Banyak penyedia layanan email menggunakan algoritma probabilistik untuk mendeteksi email yang berpotensi sebagai spam. Algoritma ini mempelajari pola dari email yang sudah dikategorikan sebagai spam atau bukan spam untuk mengidentifikasi karakteristik yang paling umum dari setiap kategori. Dengan memperkirakan probabilitas bahwa sebuah email baru adalah spam atau bukan spam berdasarkan karakteristiknya, algoritma dapat memberikan label yang sesuai.
   2) Sistem Rekomendasi: Platform rekomendasi seperti Netflix atau Spotify menggunakan algoritma probabilistik untuk memberikan rekomendasi konten kepada pengguna. Algoritma ini mempelajari preferensi pengguna berdasarkan sejarah penontonan atau pendengaran sebelumnya serta data dari pengguna dengan preferensi serupa. Dengan memperkirakan probabilitas kesukaan pengguna terhadap konten yang belum dilihat atau didengar, algoritma dapat memberikan rekomendasi yang sesuai.
   3) Prediksi Cuaca: Algoritma probabilistik juga digunakan dalam prediksi cuaca. Data cuaca yang telah dikumpulkan sebelumnya digunakan untuk mengembangkan model probabilistik yang memperkirakan kemungkinan

suhu, kelembaban, dan kondisi cuaca di masa depan. Algoritma ini memperhitungkan variasi dan ketidakpastian dalam data untuk menghasilkan perkiraan yang akurat.

4) Perdagangan Saham: Dalam perdagangan saham, algoritma probabilistik digunakan untuk mengidentifikasi pola dan tren dalam data pasar. Algoritma ini memperkirakan probabilitas pergerakan harga saham berdasarkan analisis statistik dan data historis. Dengan menggunakan informasi ini, algoritma dapat memberikan sinyal beli atau jual kepada para pedagang saham.

Dalam semua contoh ini, algoritma probabilistik menggunakan probabilitas untuk menginformasikan pengambilan keputusan atau membuat estimasi yang lebih akurat.

2. Dalam algoritma probabilistik, terdapat beberapa strategi atau pendekatan yang digunakan untuk memperkirakan solusi atau output yang benar. Beberapa di antaranya adalah:
    1) Metode Bayesian: Metode ini berdasarkan pada teorema Bayes yang menggunakan probabilitas kondisional untuk memperkirakan solusi. Dalam pendekatan ini, kita memiliki pengetahuan awal atau prior tentang solusi yang mungkin, dan kemudian memperbarui pengetahuan kita dengan mempertimbangkan data atau bukti yang baru. Contohnya adalah dalam pengenalan wajah, di mana algoritma Bayesian digunakan untuk memperkirakan kemungkinan suatu wajah yang cocok dengan citra yang diberikan.
    2) Rantai Markov: Pendekatan ini menggunakan model probabilitas untuk memodelkan transisi dari satu keadaan ke keadaan lain dalam suatu sistem. Algoritma yang menggunakan rantai Markov memperkirakan probabilitas keadaan saat ini berdasarkan keadaan sebelumnya. Contoh penerapannya adalah dalam pengenalan teks, di mana algoritma Markov digunakan untuk memperkirakan kata berikutnya dalam sebuah kalimat berdasarkan kata-kata sebelumnya.
    3) Metode Monte Carlo: Metode ini menggunakan sampel acak untuk memperkirakan solusi atau output yang benar. Dalam pendekatan ini, kita menghasilkan sejumlah besar sampel acak dari ruang solusi yang mungkin,

dan kemudian menghitung statistik dari sampel-sampel tersebut untuk memperkirakan solusi yang diinginkan. Contohnya adalah dalam estimasi nilai π menggunakan metode Monte Carlo, di mana kita menghasilkan banyak titik acak di dalam sebuah lingkaran dan menghitung rasio titik yang berada di dalam lingkaran dengan total titik yang dihasilkan.

4) Hidden Markov Model (HMM): HMM adalah model statistik yang digunakan untuk memperkirakan keadaan tersembunyi berdasarkan serangkaian pengamatan yang terlihat. Model ini memiliki keadaan tersembunyi yang tidak dapat diamati secara langsung, tetapi dapat diperkirakan melalui probabilitas pengamatan yang terjadi. Contohnya adalah dalam pengenalan suara, di mana HMM digunakan untuk memperkirakan kata atau fonem yang diucapkan berdasarkan serangkaian suara yang diamati.

Dalam semua strategi ini, algoritma probabilistik menggunakan informasi probabilitas untuk memperkirakan solusi atau output yang benar. Pendekatan ini sangat berguna dalam masalah yang kompleks atau tidak pasti di mana kita perlu menggabungkan pengetahuan awal dengan data baru untuk membuat perkiraan yang akurat.

3. Algoritma deterministik dan algoritma probabilistik berbeda dalam cara mereka memproses informasi dan menghasilkan output. Berikut adalah perbedaan antara keduanya:
    a) Algoritma Deterministik:
        - Algoritma deterministik menghasilkan output yang sama untuk setiap input yang sama. Artinya, jika inputnya tidak berubah, outputnya juga tidak akan berubah.
        - Algoritma ini mengikuti langkah-langkah yang ditentukan secara pasti dan tidak ada unsur keacakan dalam prosesnya.
        - Contoh algoritma deterministik adalah algoritma pengurutan seperti algoritma bubble sort atau algoritma pencarian seperti algoritma binary search. Dalam kedua kasus ini, setiap kali algoritma diberikan input yang sama, langkah-langkahnya akan sama dan menghasilkan output yang konsisten.
    b) Algoritma Probabilistik:

- Algoritma probabilistik menghasilkan output yang mungkin berbeda pada setiap eksekusi, bahkan jika inputnya sama.
- Algoritma ini menggunakan konsep probabilitas untuk memodelkan variasi dan ketidakpastian dalam data atau proses yang sedang dianalisis.
- Contoh algoritma probabilistik adalah algoritma pengenalan suara dengan menggunakan HMM (Hidden Markov Model). Saat algoritma ini diberikan suara yang sama, hasilnya mungkin berbeda pada setiap eksekusi. Hal ini dikarenakan algoritma memperhitungkan variasi dalam pengenalan suara dan menggunakan probabilitas untuk memperkirakan kata atau fonem yang paling mungkin berdasarkan serangkaian pengamatan.

Perbedaan utama antara kedua jenis algoritma ini adalah bahwa algoritma deterministik menghasilkan output yang konsisten dan dapat diprediksi, sedangkan algoritma probabilistik menghasilkan output yang bervariasi dan melibatkan aspek probabilitas. Algoritma probabilistik sering digunakan dalam situasi di mana variasi dan ketidakpastian merupakan faktor penting, sedangkan algoritma deterministik digunakan ketika konsistensi dan prediktabilitas output diinginkan.

4. Algoritma Naive Bayes adalah salah satu metode klasifikasi probabilistik yang sering digunakan dalam analisis teks atau analisis sentimen. Algoritma ini mengandalkan pada teorema Bayes dan mengasumsikan independensi antara fitur-fitur yang ada dalam data.

   Dalam konteks klasifikasi teks, Naive Bayes dapat digunakan untuk mengklasifikasikan dokumen atau teks ke dalam kategori atau label yang sesuai. Misalnya, dalam analisis sentimen, Naive Bayes dapat digunakan untuk mengklasifikasikan ulasan atau komentar menjadi positif, negatif, atau netral. Berikut adalah contoh penerapan Naive Bayes dalam bahasa pemrograman Python menggunakan library scikit-learn:

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

# Contoh data latih
texts = ["Ini adalah ulasan yang sangat bagus!",
         "Produk ini sangat buruk, sangat mengecewakan.",
         "Tidak terlalu bagus, tapi cukup memuaskan.",
         "Saya tidak suka dengan produk ini."]

labels = ['positif', 'negatif', 'netral', 'negatif']

# Proses pembuatan fitur teks menggunakan CountVectorizer
vectorizer = CountVectorizer()
features = vectorizer.fit_transform(texts)

# Inisialisasi model Naive Bayes
model = MultinomialNB()
```

```
# Inisialisasi model Naive Bayes
model = MultinomialNB()

# Pelatihan model
model.fit(features, labels)

# Contoh teks yang akan diklasifikasikan
new_texts = ["Produk ini luar biasa!",
             "Ulasan yang sangat jelek untuk produk ini."]

# Transformasi teks baru menjadi fitur
new_features = vectorizer.transform(new_texts)

# Klasifikasi teks baru
predictions = model.predict(new_features)

# Menampilkan hasil klasifikasi
for text, prediction in zip(new_texts, predictions):
    print(f"Teks: {text}")
    print(f"Prediksi: {prediction}\n")
```

```
# Menampilkan hasil klasifikasi
for text, prediction in zip(new_texts, predictions):
    print(f"Teks: {text}")
    print(f"Prediksi: {prediction}\n")
```

```
Teks: Produk ini luar biasa!
Prediksi: negatif

Teks: Ulasan yang sangat jelek untuk produk ini.
Prediksi: negatif
```

Dalam contoh ini, teks diwakili sebagai fitur-fitur dengan menggunakan CountVectorizer, yang mengubah teks menjadi vektor berdasarkan frekuensi kemunculan kata-kata dalam teks. Kemudian, model Naive Bayes (MultinomialNB) dilatih dengan menggunakan fitur-fitur tersebut dan label-label yang sesuai. Setelah melatih model, teks baru diubah menjadi fitur menggunakan CountVectorizer yang sama, dan kemudian model mengklasifikasikan teks baru berdasarkan fitur-fiturnya.

Pada akhirnya, hasil klasifikasi untuk teks baru ditampilkan, yaitu label atau kategori yang diprediksi oleh model.

5. Hidden Markov Model (HMM) adalah model statistik yang digunakan untuk memodelkan proses stokastik di mana keadaan yang diamati tidak dapat diamati secara langsung (keadaan tersembunyi), tetapi hanya dapat diamati melalui serangkaian pengamatan (serangkaian terlihat). HMM terdiri dari rangkaian keadaan tersembunyi yang membentuk rantai Markov dan serangkaian pengamatan yang tergantung pada keadaan tersembunyi tersebut.

   HMM memiliki dua komponen utama:
   1) Komponen keadaan tersembunyi (Hidden State): Merupakan rangkaian keadaan tersembunyi yang tidak dapat diamati secara langsung.
   2) Komponen pengamatan (Observation): Merupakan serangkaian pengamatan yang dapat diamati, tetapi tergantung pada keadaan tersembunyi.

   HMM menggunakan probabilitas transisi antar keadaan tersembunyi dan probabilitas emisi dari keadaan tersembunyi ke pengamatan untuk memodelkan hubungan antara keadaan tersembunyi dan pengamatan. HMM digunakan dalam pengenalan pola karena dapat digunakan untuk mengklasifikasikan atau memodelkan serangkaian data yang tergantung pada keadaan tersembunyi.

   Contoh penerapan HMM dalam bahasa pemrograman Python menggunakan library hmmlearn:

```
from hmmlearn import hmm
import numpy as np
```

```
# Contoh data latih
X = np.array([[0], [1], [0], [1], [0]])

# Inisialisasi model HMM
model = hmm.MultinomialHMM(n_components=2, n_iter=100)

# Pelatihan model
model.fit(X)

# Prediksi keadaan tersembunyi
hidden_states = model.predict(X)

# Menampilkan hasil prediksi
for obs, state in zip(X, hidden_states):
    print(f"Observasi: {obs[0]}, Keadaan Tersembunyi: {state}")
```

Dalam contoh ini, kita memiliki serangkaian pengamatan (X) yang terdiri dari bilangan biner (0 dan 1). Model HMM (MultinomialHMM) diinisialisasi dengan jumlah komponen keadaan tersembunyi (2) dan jumlah iterasi untuk melatih model (100). Kemudian, model dilatih menggunakan data latih (X).

Setelah melatih model, kita dapat menggunakan model untuk memprediksi keadaan tersembunyi dari serangkaian pengamatan yang diberikan. Dalam contoh ini, kita menggunakan model untuk memprediksi keadaan tersembunyi dari data latih itu sendiri. Hasil prediksi (hidden_states) menunjukkan keadaan tersembunyi yang diprediksi oleh model untuk setiap pengamatan.

Penerapan HMM dalam pengenalan pola bisa jauh lebih kompleks dengan data yang lebih besar dan kompleks, tetapi contoh di atas memberikan gambaran dasar tentang bagaimana HMM dapat digunakan untuk memodelkan dan memprediksi keadaan tersembunyi dari serangkaian pengamatan.