

Nama: Agnes Maria Anggelina

NIM: 20200801069

UTS Machine Learning

Link source github:

<https://github.com/MidnightMaria/machine-learning/tree/main/UTS>

Machine learning (ML) adalah cabang dari kecerdasan buatan yang fokus pada pengembangan teknik yang memungkinkan komputer untuk belajar dari data tanpa perlu pemrograman eksplisit. Berbeda dengan pendekatan konvensional di mana kita menulis aturan atau algoritma secara langsung untuk menyelesaikan suatu tugas, dalam machine learning, algoritma-algoritma tersebut mengoptimalkan kinerja mereka sendiri melalui pembelajaran dari data.

Ada beberapa jenis machine learning, tetapi yang paling umum adalah:

- a) Supervised Learning (Pembelajaran Terawasi): Di sini, algoritma dilatih dengan menggunakan pasangan input-output yang disebut label. Misalnya, dalam klasifikasi gambar, gambar akan menjadi input dan label yang menyatakan objek dalam gambar akan menjadi output. Algoritma belajar dari data pelatihan untuk membuat prediksi yang akurat tentang data baru.
- b) Unsupervised Learning (Pembelajaran Tanpa Pengawasan): Dalam kasus ini, algoritma bekerja dengan data yang tidak memiliki label. Tujuannya adalah untuk menemukan struktur atau pola tersembunyi dalam data tersebut. Contoh algoritma unsupervised learning termasuk klustering, di mana data dikelompokkan berdasarkan kesamaan mereka.
- c) Reinforcement Learning (Pembelajaran Penguatan): Dalam paradigma ini, agen belajar dengan berinteraksi dengan lingkungan. Agen mengambil tindakan tertentu dalam lingkungan dan menerima umpan balik (hadiah atau hukuman) berdasarkan tindakan tersebut. Tujuannya adalah untuk mengoptimalkan strategi atau kebijakan agar dapat mencapai tujuan tertentu.

1. Berikut adalah contoh penerapan implementasi machine learning dalam kehidupan sehari-hari beserta penjelasan mengenai mengapa dibutuhkan dan manfaatnya:

a) Sistem Rekomendasi

Penerapan:

Sistem rekomendasi digunakan di platform seperti Netflix, Spotify, dan Amazon. Algoritma machine learning dipakai untuk menganalisis pola perilaku pengguna, seperti riwayat penontonan atau pembelian, dan kemudian memberikan rekomendasi yang dipersonalisasi. Misalnya, Netflix menggunakan machine learning

untuk menganalisis film atau acara TV yang disukai oleh pengguna dan kemudian merekomendasikan konten serupa yang mungkin disukai.

Mengapa dibutuhkan dan Manfaatnya:

- Peningkatan Pengalaman Pengguna: Dengan menyediakan rekomendasi yang dipersonalisasi, pengguna mendapatkan konten yang lebih sesuai dengan minat dan preferensi mereka, meningkatkan kepuasan pengguna.
- Peningkatan Retensi Pengguna: Ketika pengguna merasa bahwa platform dapat memahami dan memenuhi preferensi mereka, mereka cenderung tetap menggunakan platform tersebut, meningkatkan retensi pengguna.
- Penjualan yang Lebih Tinggi: Di platform e-commerce seperti Amazon, sistem rekomendasi membantu meningkatkan penjualan dengan menyarankan produk tambahan yang relevan, menghasilkan peningkatan pendapatan.

b) Deteksi Penipuan Kartu Kredit

Penerapan:

Bank dan penyedia layanan keuangan menggunakan machine learning untuk mendeteksi aktivitas yang mencurigakan pada transaksi kartu kredit. Algoritma dipelajari menggunakan data historis tentang pola penggunaan kartu kredit yang sah dan penipuan untuk mengidentifikasi transaksi yang tidak biasa atau mencurigakan.

Mengapa dibutuhkan dan Manfaatnya:

- Pencegahan Kriminalitas Keuangan: Dengan mendeteksi penipuan kartu kredit secara cepat dan akurat, bank dapat mencegah kerugian finansial baik bagi nasabah maupun bagi diri mereka sendiri.
 - Perlindungan Pelanggan: Dengan menerapkan sistem deteksi penipuan yang efektif, bank memberikan perlindungan tambahan kepada pelanggan mereka, meningkatkan kepercayaan dan loyalitas pelanggan.
 - Efisiensi Operasional: Automatisasi proses deteksi penipuan dengan machine learning membantu bank mengurangi beban kerja manual dalam memantau transaksi secara individual, sehingga menghemat waktu dan sumber daya.
2. Berikut adalah beberapa macam taxonomi dalam penerapan machine learning:

Berdasarkan Metode Pembelajaran:

- Supervised Learning (Pembelajaran Terawasi): Algoritma belajar dari data yang memiliki label. Contohnya adalah klasifikasi (memisahkan data ke dalam kategori) dan regresi (memperkirakan nilai berkelanjutan).

- Unsupervised Learning (Pembelajaran Tanpa Pengawasan): Algoritma belajar dari data yang tidak memiliki label. Contohnya termasuk klustering (mengelompokkan data berdasarkan kesamaan) dan reduksi dimensi (mengurangi jumlah fitur dalam data).

- Reinforcement Learning (Pembelajaran Penguatan): Algoritma belajar dengan berinteraksi dengan lingkungan dan menerima umpan balik dalam bentuk hadiah atau hukuman. Contohnya adalah pembelajaran dalam permainan video atau pengaturan otomatis.

Berdasarkan Pendekatan Algoritma:

- Algoritma Linear: Algoritma yang bergantung pada hubungan linier antara variabel input dan output.

- Algoritma Non-linear: Algoritma yang dapat menangani hubungan yang kompleks dan non-linear antara variabel input dan output, seperti Support Vector Machines (SVM) atau Neural Networks (jaringan saraf).

- Algoritma Berbasis Probabilitas: Algoritma yang mengambil pendekatan berbasis probabilitas, seperti Naive Bayes atau Bayesian Networks.

Berdasarkan Tujuan Aplikasi:

- Klasifikasi: Mengelompokkan data ke dalam kategori atau kelas tertentu, seperti mengklasifikasikan email sebagai spam atau non-spam.

- Regresi: Memperkirakan nilai berkelanjutan berdasarkan data input, seperti memprediksi harga rumah berdasarkan fitur-fitur tertentu.

- Klustering: Mengelompokkan data ke dalam kelompok yang memiliki kesamaan, seperti mengelompokkan pelanggan berdasarkan pola pembelian mereka.

- Pencarian Pola (Pattern Recognition): Mengidentifikasi pola atau struktur tersembunyi dalam data, seperti mengidentifikasi pola dalam data medis untuk diagnosis penyakit.

4.

Studi Kasus 1

```
import pandas as pd

# Membuat DataFrame dengan data yang diberikan
data = {
    "HARI": ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"],
    "DATANG": [2, 3, 4, 1, 2, 5, 2],
    "BIAYA": [30000*2, 35000*3, 25000*4, 15000*1, 20000*2, 30000*5, 35000*2],
    "MAHASISWA": ["Ani", "Budi", "Jono", "Lono", "Joni", "Ani", "Budi"]
}

df = pd.DataFrame(data)

# a) Rata-rata mahasiswa datang pada minggu ini
rata_rata_datang = df["DATANG"].mean()
```

```

# b) Kapan biaya tertinggi terjadi?
biaya_tertinggi = df[df["BIAYA"] == df["BIAYA"].max()]["HARI"].values[0]

# c) Hari apa biaya lebih dari 110000?
hari_biaya_lebih_110000 = df[df["BIAYA"] > 110000]["HARI"].values

# d) Siapa yang paling banyak datang ke kampus?
mahasiswa_paling_banyak_datang = df["MAHASISWA"].value_counts().idxmax()

# e) Siapa yang datang pada hari Minggu?
mahasiswa_datang_minggu = df[df["HARI"] == "Minggu"]["MAHASISWA"].values

# f) Biaya tertinggi dan terendah
biaya_tertinggi = df["BIAYA"].max()
biaya_terendah = df["BIAYA"].min()

# g) Frekuensi datang tertinggi dan terendah
frekuensi_datang_tertinggi = df["DATANG"].max()
frekuensi_datang_terendah = df["DATANG"].min()

# Menampilkan hasil
print("a) Rata-rata mahasiswa datang pada minggu ini:", rata_rata_datang)
print("b) Kapan biaya tertinggi terjadi?:", biaya_tertinggi)
print("c) Hari apa biaya lebih dari 110000?:", hari_biaya_lebih_110000)
print("d) Siapa yang paling banyak datang ke kampus?:", mahasiswa_paling_banyak_datang)
print("e) Siapa yang datang pada hari Minggu?:", mahasiswa_datang_minggu)
print("f) Biaya tertinggi:", biaya_tertinggi)
print("    Biaya terendah:", biaya_terendah)
print("g) Frekuensi datang tertinggi:", frekuensi_datang_tertinggi)
print("    Frekuensi datang terendah:", frekuensi_datang_terendah)

```

```

a) Rata-rata mahasiswa datang pada minggu ini: 2.7142857142857144
b) Kapan biaya tertinggi terjadi?: 150000
c) Hari apa biaya lebih dari 110000?: ['Sabtu']
d) Siapa yang paling banyak datang ke kampus?: Ani
e) Siapa yang datang pada hari Minggu?: ['Budi']
f) Biaya tertinggi: 150000
    Biaya terendah: 15000
g) Frekuensi datang tertinggi: 5
    Frekuensi datang terendah: 1

```

Studi Kasus 2

```

import pandas as pd
import matplotlib.pyplot as plt

# Membuat data frame dengan informasi mahasiswa
fakultas = ["Bisnis", "D3 Perhotelan", "ICT", "Ilmu Komunikasi", "Seni dan Desain"]
jumlah_mahasiswa = [260, 28, 284, 465, 735]

```

```

akreditasi = ["A", "A", "B", "A", "A"]
info_mahasiswa = pd.DataFrame({"fakultas": fakultas, "jumlah_mahasiswa": jumlah_mahasiswa,
                                "akreditasi": akreditasi})

# Menampilkan data frame
print(info_mahasiswa)

# Membuat warna untuk setiap fakultas
colors = ['orange', 'yellow', 'green', 'blue', 'purple']

# Membuat grafik batang jumlah mahasiswa per fakultas
plt.bar(info_mahasiswa["fakultas"], info_mahasiswa["jumlah_mahasiswa"], color=colors)
plt.title("Jumlah Mahasiswa per Fakultas")
plt.xlabel("Fakultas")
plt.ylabel("Jumlah Mahasiswa")
plt.xticks(rotation=45)
plt.show()

```

	fakultas	jumlah_mahasiswa	akreditasi
0	Bisnis	260	A
1	D3 Perhotelan	28	A
2	ICT	284	B
3	Ilmu Komunikasi	465	A
4	Seni dan Desain	735	A

