

Nama: Wiryawan Pranawadigda

Nim: 20200801050

Link Github: <https://github.com/Mone164/UTS-Machine-Learning>

UTS Machine Learning

1. Machine Learning (ML) adalah cabang dari kecerdasan buatan (Artificial Intelligence, AI) yang fokus pada pengembangan algoritma dan teknik yang memungkinkan komputer untuk belajar dari dan membuat prediksi atau keputusan berdasarkan data. Dengan kata lain, Machine Learning memungkinkan sistem untuk secara otomatis belajar dan meningkatkan kinerja mereka dari pengalaman tanpa harus diprogram secara eksplisit untuk tugas tertentu.

2. Machine Learning (ML) memiliki berbagai aplikasi dalam kehidupan sehari-hari yang membawa manfaat besar dalam meningkatkan efisiensi, kenyamanan, dan kualitas hidup. Berikut beberapa contoh penerapan ML dan penjelasan mengapa membutuhkannya serta manfaatnya:

a. Rekomendasi Produk dan Konten

Contoh: Platform seperti Amazon, Netflix, dan Spotify menggunakan algoritma ML untuk merekomendasikan produk, film, dan musik berdasarkan preferensi pengguna.

Mengapa Diperlukan: Dengan banyaknya pilihan yang tersedia, pengguna dapat kesulitan menemukan apa yang mereka sukai. Algoritma rekomendasi membantu menyaring konten dan produk yang paling relevan.

Manfaat: Meningkatkan pengalaman pengguna dengan menyarankan konten yang sesuai dengan selera mereka, yang dapat meningkatkan kepuasan dan keterlibatan pengguna serta meningkatkan penjualan untuk perusahaan.

b. Personal Assistant dan Chatbots

Contoh: Siri, Google Assistant, dan Alexa menggunakan ML untuk memahami dan merespons perintah suara pengguna.

Mengapa Diperlukan: Pengguna sering membutuhkan cara cepat dan efisien untuk mengakses informasi atau melakukan tugas-tugas tertentu.

Manfaat: Asisten virtual ini dapat membantu dengan berbagai tugas sehari-hari, seperti mengatur jadwal, mengirim pesan, mencari informasi di internet, dan banyak lagi, yang meningkatkan efisiensi dan kenyamanan.

c. Diagnosa Medis

Contoh: Algoritma ML digunakan dalam radiologi untuk mendeteksi penyakit seperti kanker dari gambar medis, atau dalam analisis data genetik untuk prediksi penyakit.

Mengapa Diperlukan: Diagnosis medis yang cepat dan akurat sangat penting untuk pengobatan dini dan peningkatan peluang kesembuhan.

Manfaat: Meningkatkan akurasi diagnosis, mempercepat waktu deteksi penyakit, dan membantu dokter dalam membuat keputusan pengobatan yang lebih baik.

3. Taxonomi dalam penerapan Machine Learning (ML) mengacu pada berbagai klasifikasi atau kategori yang digunakan untuk mengelompokkan jenis-jenis pembelajaran mesin berdasarkan metode dan tujuannya. Berikut adalah macam-macam taxonomi dalam penerapan Machine Learning:

1. Berdasarkan Jenis Pembelajaran

a. Supervised Learning (Pembelajaran Terawasi)

Definisi: Model dilatih menggunakan data yang diberi label, di mana setiap input memiliki output yang diinginkan.

Contoh Algoritma: Linear Regression, Logistic Regression, Decision Trees, Support Vector Machines, Neural Networks.

Aplikasi: Klasifikasi email sebagai spam atau bukan, prediksi harga rumah, pengenalan wajah.

b. Unsupervised Learning (Pembelajaran Tak Terawasi)

Definisi: Model dilatih menggunakan data yang tidak diberi label. Model mencoba menemukan pola atau struktur tersembunyi dalam data.

Contoh Algoritma: K-Means Clustering, Hierarchical Clustering, Principal Component Analysis (PCA), Autoencoders.

Aplikasi: Segmentasi pelanggan, pengelompokan dokumen, deteksi anomali.

2. Berdasarkan Jenis Output

a. Regression (Regresi)

Definisi: Prediksi nilai kontinu.

Contoh: Prediksi harga rumah, prediksi suhu, prediksi penjualan.

b. Classification (Klasifikasi)

Definisi: Pengelompokan input ke dalam kategori diskrit.

Contoh: Klasifikasi email sebagai spam atau bukan, klasifikasi gambar kucing atau anjing, deteksi penyakit.

3. Berdasarkan Teknik

a. Instance-based Learning

Definisi: Model menyimpan contoh-contoh pelatihan dan membuat prediksi berdasarkan kemiripan baru.

Contoh Algoritma: K-Nearest Neighbors (KNN).

Aplikasi: Pengenalan pola, klasifikasi teks.

b. Model-based Learning

Definisi: Model membuat abstraksi atau generalisasi dari data pelatihan untuk membuat prediksi.

Contoh Algoritma: Linear Regression, Support Vector Machines, Neural Networks.

Aplikasi: Regresi, klasifikasi.

Studi Kasus:

1. Untuk Mencari hasil digunakan code phyton seperti berikut:

```
import pandas as pd
```

```
# Data mahasiswa dan biaya
```

```
mahasiswa = {  
    "Ani": 30000,  
    "Budi": 35000,  
    "Joni": 20000,  
    "Jono": 25000,  
    "Lono": 15000  
}
```

```
# Data kedatangan
```

```
data_kedatangan = {  
    "Senin": {"jumlah": 2, "mahasiswa": ["Ani"], "biaya": 30000 * 2},  
    "Selasa": {"jumlah": 3, "mahasiswa": ["Budi"], "biaya": 35000 * 3},  
    "Rabu": {"jumlah": 4, "mahasiswa": ["Jono"], "biaya": 25000 * 4},  
    "Kamis": {"jumlah": 1, "mahasiswa": ["Lono"], "biaya": 15000 * 1},  
    "Jumat": {"jumlah": 2, "mahasiswa": ["Joni"], "biaya": 20000 * 2},  
    "Sabtu": {"jumlah": 5, "mahasiswa": ["Ani"], "biaya": 30000 * 5},  
    "Minggu": {"jumlah": 2, "mahasiswa": ["Budi"], "biaya": 35000 * 2}  
}
```

```
# Membuat DataFrame
```

```
data = []  
for hari, info in data_kedatangan.items():  
    data.append([hari, info["jumlah"], ", ".join(info["mahasiswa"]), info["biaya"]])
```

```
df = pd.DataFrame(data, columns=["Hari", "Jumlah", "Mahasiswa", "Biaya"])  
print(df)
```

```
# a) Berapa rata-rata mahasiswa datang pada minggu ini?
```

```
rata_rata_datang = df["Jumlah"].mean()  
print("Rata-rata mahasiswa datang pada minggu ini:", rata_rata_datang)
```

```
# b) Kapan biaya tertinggi terjadi?
```

```
biaya_tertinggi = df.loc[df["Biaya"].idxmax()]  
print("Biaya tertinggi terjadi pada hari:", biaya_tertinggi["Hari"])
```

```
# c) Hari apa biaya lebih dari 110000?
```

```
biaya_lebih_110k = df[df["Biaya"] > 110000]["Hari"].tolist()  
print("Hari dengan biaya lebih dari 110000:", biaya_lebih_110k)
```

```
# d) Siapa yang paling banyak datang ke kampus?
# Menghitung frekuensi kedatangan setiap mahasiswa
mahasiswa_counts = df["Mahasiswa"].str.get_dummies(sep=', ').sum()
mahasiswa_tersering_datang = mahasiswa_counts.idxmax()
print("Mahasiswa yang paling banyak datang ke kampus:", mahasiswa_tersering_datang)
```

```
# e) Siapa yang datang pada hari Minggu?
datang_minggu = df[df["Hari"] == "Minggu"]["Mahasiswa"].unique()
print("Mahasiswa yang datang pada hari Minggu:", datang_minggu)
```

```
# f) Berapa biaya tertinggi dan terendah?
biaya_tertinggi = df["Biaya"].max()
biaya_terendah = df["Biaya"].min()
print("Biaya tertinggi:", biaya_tertinggi)
print("Biaya terendah:", biaya_terendah)
```

```
# g) Berapa frekuensi datang tertinggi dan terendah?
frekuensi_tertinggi = df["Jumlah"].max()
frekuensi_terendah = df["Jumlah"].min()
print("Frekuensi datang tertinggi:", frekuensi_tertinggi)
print("Frekuensi datang terendah:", frekuensi_terendah)
```

a. Rata-rata mahasiswa datang pada minggu ini: 2.7142857142857144

b. Biaya tertinggi terjadi pada hari: Sabtu

c. Hari dengan biaya lebih dari 110000: ['Selasa', 'Sabtu']

d. Mahasiswa yang paling banyak datang ke kampus: Ani

e. Mahasiswa yang datang pada hari Minggu: ['Budi']

f. Biaya tertinggi: 150000

Biaya terendah: 15000

g. Frekuensi datang tertinggi: 5

Frekuensi datang terendah: 1

```
2. import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Data
```

```
fakultas = ["Bisnis", "D3 Perhotelan", "ICT", "Ilmu Komunikasi", "Seni dan Desain"]
```

```
jumlah_mahasiswa = [260, 28, 284, 465, 735]
akreditasi = ["A", "A", "B", "A", "A"]
```

```
# Membuat DataFrame
info_mahasiswa = pd.DataFrame({
    'fakultas': fakultas,
    'jumlah_mahasiswa': jumlah_mahasiswa,
    'akreditasi': akreditasi
})
```

```
# Menampilkan DataFrame
print(info_mahasiswa)
```

```
# Visualisasi
plt.figure(figsize=(10, 6))
sns.barplot(x='fakultas', y='jumlah_mahasiswa', data=info_mahasiswa, palette="viridis")
```

```
# Menambahkan judul dan label
plt.title('Jumlah Mahasiswa per Fakultas')
plt.xlabel('Fakultas')
plt.ylabel('Jumlah Mahasiswa')
plt.xticks(rotation=45)
```

```
# Menampilkan plot
plt.show()
```