Logistic Regression

Mohammad Febryan Khamim

1 Pengantar Logistic Regression

Logistic Regression adalah algoritma *supervised learning* untuk permasalahan klasifikasi. Regresi ini menggunakan klasifikasi *binary* sehingga menghasilkan output *binary*. ***Logistic Regression*** menggunakan **fungsi sigmoid** untuk meng-*convert* input menjadi nilai probabilitas.

2 Linear Regression vs Logistic Regression

Terdapat beberapa perbedaan antara regresi linear dan regresi logistik, di antaranya sebagai berikut.

Regresi Linear	**Regresi Logistik**
Prediksi nilai kontinu	Prediksi kelas kategorikal
Menggunakan *best-fit line*	Menggunakan kurva sigmoid
Menyelesaikan masalah regresi	Menyelesaikan masalah klasifikasi

3 Tipe-Tipe Logistic Regression

Terdapat beberapa tipe Logistic Regression berdasarkan variabel dependennya, di antaranya:

- **Binomial Logistic Regression (Regresi Logistik Binomial)**
  Tipe ini digunakan ketika variabel dependen hanya memiliki dua kategori kemungkinan (biner).
  - *Contoh:* Ya atau Tidak, Lulus atau Gagal, 0 atau 1.
- **Multinomial Logistic Regression (Regresi Logistik Multinomial)**
  Tipe ini digunakan ketika variabel dependen memiliki tiga atau lebih kategori kemungkinan yang tidak memiliki urutan atau peringkat tertentu.
  - *Contoh:* Klasifikasi jenis hewan (Kucing, Anjing, Domba).
- **Ordinal Logistic Regression (Regresi Logistik Ordinal)**
  Tipe ini digunakan ketika variabel dependen memiliki tiga atau lebih kategori yang memiliki urutan alami atau tingkatan (peringkat).
  - *Contoh:* Penilaian atau rating (Low, Medium, High).

4 Asumsi Regresi Logistik

1. Observasi independen → Tak ada korelasi antar sampel
2. Variabel binary dependen → Variabel harus biner, gunakan **softmax**
3. Hubungan linearitas antara variabel & log → Transformasi fitur
4. Tak ada *outlier*
5. Ukuran sampel besar

5 Apa Itu Fungsi Sigmoid?

- Sigmoid merupakan fungsi untuk convert output mentah menjadi nilai probabilitas antara 0–1.
- Fungsi ini mengonversi sembarang angka menjadi antara 0–1 yang membentuk kurva "S".
- Dalam **Logistic Regression**, digunakan *threshold*, umumnya 0,5.
  - Jika sigmoid lebih dari atau sama dengan *threshold* → anggap 1
  - Jika di bawah 0 → anggap 0

6 Konsep Matematis Logistic Regression

Berikut adalah konsep matematis dalam algoritma Logistic Regression:

1. Anggap terdapat sebuah fitur input yang direpresentasikan dalam matriks dan variabel dependen Y yang bernilai biner 0 atau 1.

$$\begin{bmatrix} x_{11} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

dan

$$Y = \begin{cases} 0, & \text{jika class 1,} \\ 1, & \text{jika class 2.} \end{cases}$$

2. Dari input tersebut, maka berlaku kombinasi linear:

$$z = \left(\sum_{i=1}^{n} w_i x_i \right) + b$$

dengan:

- $\sum_{i=1}^{n} w_i x_i$ = bobot / koefisien
- z = skor linear

- b = intercept
- x_i = fitur input

3. Mengubah dari nilai kontinu z menjadi probabilitas antara 0 hingga 1 yang dapat digunakan untuk prediksi kelas

   (*) Gunakan fungsi Sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

   Sehingga, diperoleh grafik '**S-*Shape***' dengan:

   - $\sigma(z) = 1$ saat $z \to \infty$
   - $\sigma(z) = 0$ saat $z \to -\infty$
   - $\sigma(z)$ selalu di antara 0 hingga 1

   dengan probabilitas kelas dalam ***Logistic Regression*** dapat dihitung:

   Persamaan Logistic Regression dan Peluang

   Berikut adalah penurunan hubungan antara *odds* dan probabilitas:

   (a) **Definisi Odds:**

$$\frac{P(x)}{1 - P(x)} = e^z$$

   (b) **Log-Odds (Logit):**

$$\begin{aligned} \log\left(\frac{P(x)}{1 - P(x)}\right) &= \log e^z \\ \log\left(\frac{P(x)}{1 - P(x)}\right) &= z \\ &= w \cdot X + b \end{aligned}$$

   (c) **Penurunan Menuju Fungsi Sigmoid:**

$$\begin{aligned} \frac{P(x)}{1 - P(x)} &= e^{w \cdot X + b} \\ P(x) &= (1 - P(x))e^{w \cdot X + b} \\ P(x) &= e^{w \cdot X + b} - P(x)e^{w \cdot X + b} \\ P(x)\left(1 + e^{w \cdot X + b}\right) &= e^{w \cdot X + b} \end{aligned}$$

   Sehingga diperoleh fungsi probabilitas:

$$P(x) = \frac{e^{w \cdot X + b}}{1 + e^{w \cdot X + b}} \quad \text{atau} \quad P(x) = \frac{1}{1 + e^{-(w \cdot X + b)}}$$

7 Penerapan pada Program Python

7.1 Import Libraries yang Dibutuhkan

```
import pandas as pd
import numpy as np


from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

7.2 Membaca File CSV dan Informasi pada Data

```
data = pd.read_csv("data.csv")


print(data.head())
print(data.info())
print(data.describe())
```

7.3 Prapemrosesan Data

```
# Cek missing values
print(data.isnull().sum())


# Mengisi missing values (contoh: numerik dengan mean)
for col in data.select_dtypes(include=np.number).columns:
data[col].fillna(data[col].mean(), inplace=True)


# Drop fitur yang tidak dibutuhkan
data.drop(columns=['ID'], inplace=True, errors='ignore')
```

7.4 Konversi Data Kategorikal

```
label_encoders = {}


for col in data.select_dtypes(include='object').columns:
le = LabelEncoder()
data[col] = le.fit_transform(data[col])
label_encoders[col] = le
```

7.5 Diperoleh Fitur yang Akan Dipakai

```
X = data.drop(columns=['Target'])
y = data['Target']
```

```
print(X.head())
print(y.head())
```

7.6 Konstruksi Model

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=42
)


# Standarisasi fitur
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


# Inisialisasi model
model = LogisticRegression()
```

7.7 *Running* Model

```
model.fit(X_train, y_train)


y_pred = model.predict(X_test)
```

7.8 Evaluasi Model

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)


print("\nClassification Report:")
print(classification_report(y_test, y_pred))


print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

8 Full Program Logistic Regression

Listing 1: Program Logistic Regression Python

```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplorlib inline


# PRA-PEMROSESAN DATA

# Membaca data
data = pd.read_csv(r'E:\Perkuliahan\Karier\13-Logistic-Regression\
   advertising.csv')

# Menampilkan Informasi Dasar tentang Data
data.head()
data.info()
data.describe()

# EXPLORATORY DATA ANALYSIS (EDA)

# Membuat Histogram untuk Age
sns.set_style('whitegrid')
data['Age'].hist(bins=30)
plt.xlabel('Age')

# Membuat Jointplot antara Age dan Area Income
sns.jointplot(x='Age', y='Area Income', data=data)

# Membuat Jointplot antara Daily Time Spent on Site dan Age
sns.jointplot(x='Age',y='Daily Time Spent on Site',data=ad_data,
   color='red',kind='kde')

# Membuat jointplot antara Daily Time Spent on Site dan Daily
   Internet Usage
sns.jointplot(x='Daily Time Spent on Site',y='Daily Internet Usage'
   ,data=ad_data,color='green')

# Membuat pairplot dengan hue berdasarkan Clicked on Ad
# Clicked on Ad adalah variabel target
sns.pairplot(data,hue='Clicked on Ad',palette='bwr')

# PELATIHAN MODEL LOGISTIC REGRESSION

# Memisahkan data train dan data test
from sklearn.model_selection import train_test_split
X = data[['Daily Time Spent on Site', 'Age', 'Area Income','Daily
   Internet Usage', 'Male']]
```

```
y = data['Clicked on Ad']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.33, random_state=42)

# Melatih model Logistic Regression
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)

# MELAKUKAN PREDIKSI
predictions = logmodel.predict(X_test)

# Evaluasi Model
from sklearn.metrics import classification_report, confusion_matrix
print(classification_report(y_test,predictions))
```