Regresi Logistik

Regresi logistik adalah model statistik yang digunakan untuk memprediksi probabilitas suatu peristiwa terjadi berdasarkan satu atau lebih variabel independen. Regresi logistik sering digunakan untuk masalah klasifikasi biner, dimana variabel dependen (target) hanya memiliki dua kemungkinan nilai, misalnya “ya/tidak”, “sukses/gagal”, “sehat/sakit”, dan sebagainya.

# Perbedaan dengan Regresi Linear

Meskipun namanya mirip, regresi logistik berbeda dengan regresi linear. Perbedaan utama terletak pada jenis variabel dependennya. Pada regresi linear, variabel dependen adalah numerik dan kontinu, sedangkan pada regresi logistik, variabel dependen adalah kategorikal biner.

# Fungsi Logistik

Regresi logistik menggunakan fungsi logistik (sigmoid) untuk memetakan nilai prediksi linear ke probabilitas. Fungsi logistik memiliki bentuk seperti huruf “S” dan membatasi nilai probabilitas antara 0 dan 1.

# Interpretasi Koefisien Regresi

Koefisien regresi dalam logistik mengindikasikan perubahan log-odds dari variabel dependen ketika variabel independen meningkat sebesar satu unit. Log-odds adalah logaritma natural dari odds, yaitu rasio antara probabilitas suatu peristiwa terjadi dan probabilitas peristiwa tersebut tidak terjadi.

# Evaluasi Model Regresi Logistik

Matrik evaluasi yang umum digunakan untuk regresi logistik sama dengan metrik untuk model klasifikasi lainnya, seperti:

* Akurasi (Accuracy)
* Presisi (Precision)
* Recall (Sensitivity)
* F1-Score
* ROC AUC (Area Under the Receiver Operating Characteristic Curve)

# Penerapan Regresi Logistik dengan Scikit-Learn

Scikit-learn menyediakan modul LogisticRegression untuk membangun model regresi logistik.

**Tugas Hari 18:**

Anda akan bekerja dengan dataset “Social Network Ads”. Dataset ini berisi informasi tentang pengguna media sosial, termasuk usia, estimasi gaji, dan apakah mereka membeli suatu produk (Purchased) atau tidak.

**Tugas**

1. **Memuat dan Visualisasi Data:**

* Unduh dataset “Social Network Ads” dari Kaggle.
* Gunakan Pandas untuk membaca dataset.
* Buat scatter plot antara Age dan EstimatedSalary, dengan warna titik berbeda untuk setiap nilai Purchased.

1. **Membuat Model Regresi Logistik:**

* Gunakan LogistikRegression dari scikit-learn untuk membuat model.
* Gunakan Age dan EstimatedSalary sebagai fitur dan Purchased sebagai target.
* Bagi data menjadi data latih dan data uji dengan proporsi 80:20.
* Latih model pada data latih.

1. **Prediksi dan Evaluasi:**

* Gunakan model untuk memprediksi label kelas pada data uji.
* Hitung akurasi, presisi, recall, dan F1-score dari model pada data uji.

**Contoh Kode (Scikit-Learn dan Matplotlib)**

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

from sklearn.preprocessing import StandardScaler

# ... (kode untuk membaca dan mempersiapkan dataset Social Network Ads)

# Scatter plot

plt.figure(figsize=(8, 6))

plt.scatter(X\_train[:, 0], X\_train[:, 1], c=y\_train, cmap='coolwarm')

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.title('Scatter Plot of Age vs. Estimated Salary by Purchased')

plt.show()

# Model regresi logistik

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Prediksi dan evaluasi

y\_pred = model.predict(X\_test)

# ... (hitung akurasi, presisi, recall, dan F1-score)

**Catatan:**

* Lakukan standardisasi fitur (scaling) sebelum melatih model. Hal ini penting karena perbedaan skala antar fitur dapat mempengaruhi kinerja model.

**Selamat Mengerjakan Tugas 😀**

**Tugas**

1. **Memuat dan Visualisasi Data:**

**# Scatter Plot (Visualisasi setelah Prediksi)**

**plt.figure(figsize=(8, 6))**

**plt.scatter(X\_test[:, 0], X\_test[:, 1], c=y\_pred, cmap='coolwarm') # Menggunakan y\_pred**

**plt.xlabel('Age')**

**plt.ylabel('Estimated Salary')**

**plt.title('Scatter Plot of Age vs. Estimated Salary by Purchased (Predicted)')**

**plt.show()**

1. **membuat Model Regresi Logistik:**

**# Membuat Model Regresi Logistik**

**model = LogisticRegression()**

**# Melatih Model**

**model.fit(X\_train, y\_train)**

1. **Prediksi dan Evaluasi:**

**# Prediksi pada Data Uji**

**y\_pred = model.predict(X\_test)**

**# Evaluasi Model**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**precision = precision\_score(y\_test, y\_pred)**

**recall = recall\_score(y\_test, y\_pred)**

**f1 = f1\_score(y\_test, y\_pred)**

**print(f'Akurasi: {accuracy:.2f}')**

**print(f'Presisi: {precision:.2f}')**

**print(f'Recall: {recall:.2f}')**

**print(f'F1-Score: {f1:.2f}')**

**CONTOH VISUALISASI BATAS KEPUTUSAN!**

# ... (kode sebelumnya)

# Membuat meshgrid untuk visualisasi batas keputusan

x\_min, x\_max = X\_test[:, 0].min() - 1, X\_test[:, 0].max() + 1

y\_min, y\_max = X\_test[:, 1].min() - 1, X\_test[:, 1].max() + 1

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, 0.01),

np.arange(y\_min, y\_max, 0.01))

# Prediksi probabilitas untuk setiap titik pada meshgrid

Z = model.predict(np.c\_[xx.ravel(), yy.ravel()])

Z = Z.reshape(xx.shape)

# Plot scatter plot dengan batas keputusan

plt.contourf(xx, yy, Z, alpha=0.4, cmap='coolwarm') # Batas keputusan

plt.scatter(X\_test[:, 0], X\_test[:, 1], c=y\_test, cmap='coolwarm') # Data asli dengan label

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.title('Scatter Plot of Age vs. Estimated Salary with Decision Boundary')

plt.show()