1. Konsep Regresi Logistik

Konsep Regresi Logistik

- Tujuan: Regresi logistik digunakan untuk memprediksi probabilitas hasil biner (yaitu, dua kemungkinan hasil seperti “ya” atau “tidak”, spam, atau “bukan spam”.
- Regresi logistik dapat memprediksi apakah pasien menderita penyakit berdasarkan berbagai faktor diagnostik.
- Fungsi Logistik (Fungsi Sigmoid): Model regresi logistik menggunakan fungsi logistik untuk memodelkan variabel dependen biner. Fungsinya adalah:

$$\text{Fungsi Sigmoid:}\sigma(z) = \frac{1}{1 + e^{-z}}$$

Di mana z adalah kombinasi linier dari variabel input (prediktor).

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Dengan asumsi kita memiliki kumpulan data dengan fitur X dan target y, Anda ingin mengoptimalkan model regresi logistik:

$$\hat{y} = \frac{1}{1 + e^{-z}}$$

di mana:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

maka

$$\hat{y} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

Rumus Fungs Cost

Fungsi cost dalam regresi logistik, yang ingin akan diminimalkan, adalah negatif log-likelihood:

$$J(\beta) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Penjelasan

Untuk menurunkan fungsi biaya $J(\beta)$ terhadap β_0, β_1, dan β_2, kita dapat menggunakan turunan parsial dari fungsi biaya tersebut. Fungsi biaya $J(\beta)$ untuk regresi logistik diberikan oleh:

$$J(\beta) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

di mana $\hat{y}_i$ adalah probabilitas prediksi untuk observasi ke-i yang didefinisikan sebagai:

$$\hat{y}_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2})}}$$

Mari kita turunkan $J(\beta)$ terhadap masing-masing parameter.

Turunan Terhadap β_0

1. **Hitung Turunan $\hat{y}_i$ Terhadap β_0:**

$$\frac{\partial \hat{y}_i}{\partial \beta_0} = \hat{y}_i(1 - \hat{y}_i)$$

2. **Turunkan Fungsi Biaya $J(\beta)$:**

$$\frac{\partial J(\beta)}{\partial \beta_0} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y_i}{\hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \beta_0} - \frac{(1 - y_i)}{1 - \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \beta_0}\right]$$

Substitusi turunan $\hat{y}_i$:

$$\frac{\partial J(\beta)}{\partial \beta_0} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)$$

Turunan Terhadap β_1

1. **Hitung Turunan $\hat{y}_i$ Terhadap β_1:**

$$\frac{\partial \hat{y}_i}{\partial \beta_1} = \hat{y}_i(1 - \hat{y}_i) \cdot x_{i1}$$

2. **Turunkan Fungsi Biaya $J(\beta)$:**

$$\frac{\partial J(\beta)}{\partial \beta_1} = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y_i}{\hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \beta_1} - \frac{(1 - y_i)}{1 - \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \beta_1}\right]$$

Substitusi turunan $\hat{y}_i$:

$$\frac{\partial J(\beta)}{\partial \beta_1} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i) \cdot x_{i1}$$

Turunan Terhadap β_2

1. **Hitung Turunan $\hat{y}_i$ Terhadap β_2:**

$$\frac{\partial \hat{y}_i}{\partial \beta_2} = \hat{y}_i(1 - \hat{y}_i) \cdot x_{i2}$$

2. **Turunkan Fungsi Biaya** $J(\beta)$**:**

$$\frac{\partial J(\beta)}{\partial \beta_2} = -\frac{1}{m} \sum_{i=1}^{m} \left[\frac{y_i}{\hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \beta_2} - \frac{(1 - y_i)}{1 - \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \beta_2} \right]$$

Substitusi turunan $\hat{y}_i$:

$$\frac{\partial J(\beta)}{\partial \beta_2} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) \cdot x_{i2}$$

Kesimpulan

Turunan dari fungsi biaya $J(\beta)$ terhadap parameter β_0, β_1, dan β_2 adalah:

$$\frac{\partial J(\beta)}{\partial \beta_0} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)$$

$$\frac{\partial J(\beta)}{\partial \beta_1} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) \cdot x_{i1}$$

$$\frac{\partial J(\beta)}{\partial \beta_2} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) \cdot x_{i2}$$

Gunakan turunan-turunan ini dalam algoritma gradient descent untuk memperbarui β_0, β_1, dan β_2.

Terapkan Gradient Descent

Implementasi gradient descent untuk memperbarui β secara berulang hingga konvergensi: 1. Inisialisasi β: Mulailah dengan β yang ditetapkan ke nol atau nilai acak kecil 2. Pilih learning rate α: Angka positif kecil seperti 0,01. 3. Perbarui β menggunakan gradien

$$\beta_j := \beta_j - \alpha \cdot \frac{\partial J(\beta)}{\partial \beta_j}$$

4. Ulangi: Terus perbarui β hingga fungsi biaya $J(\beta)$ konvergen (yaitu, perubahan berada di bawah ambang batas yang ditetapkan) atau hingga jumlah iterasi maksimum tercapai.

Contoh Menghitung Gradien untuk β_1

Gradien fungsi cost terhadap β_1 adalah:

$$\frac{\partial J(\beta)}{\partial \beta_1} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) x_{i1}$$

di mana x_{i1} adalah nilai fitur pertama untuk observasi ke-i, dan $\hat{y}_i$ adalah probabilitas yang diprediksi untuk observasi ke-i.

Contoh Perbarui β_1 Menggunakan Gradient Descent

Perbarui β_1 secara berulang:

1. Inisialisasi β_1 :Mulai dengan nilai seperti 0 atau angka acak kecil.
2. Pilih learning rate α :Misalnya, 0,01.
3. Perbarui β_1

$$\beta_1 := \beta_1 - \alpha \cdot \frac{\partial J(\beta)}{\partial \beta_1}$$

4. Ulangi: Terus perbarui β_1 dan koefisien lainnya hingga konvergensi