Logistic Regression

Adinda Putri - 13523071

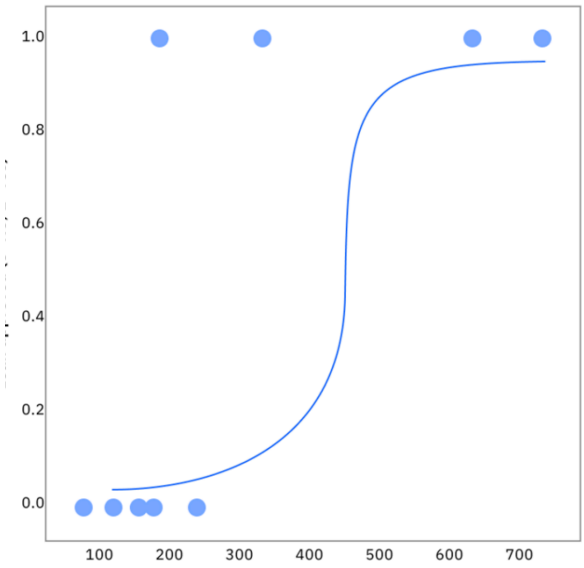
Logistic regression merupakan model linear yang mencari hubungan antara predictor variable (independent variable) dengan output variable (target atau dependent variable). Model ini berbeda dengan linear regression. Linear regression digunakan ketika output berupa nilai kontinu. Sementara logistic regression digunakan ketika output bersifat kategorikal seperti apakah hari ini akan hujan atau tidak, atau apakah terdapat penyakit tertentu pada seorang pasien atau tidak.

Cara Kerja

Hubungan antara independent variable dan output pada logistic regression secara matematis adalah sebagai berikut:

$$Y = P\left(x \right) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

dimana Y merupakan probabilitas terjadinya suatu output berdasarkan x. Nilai Y terbatas pada interval [0, 1] sebagaimana nilai probabilitas semestinya. Model menggunakan persamaan ini untuk memprediksi dan mengklasifikasikan apakah output tersebut terjadi atau tidak. Nilai output pada persamaan di atas berkisar antara 0-1, lalu berdasarkan nilai threshold, biasanya 0.5, output tersebut diklasifikasikan menjadi salah satu di antara 0 (tidak terjadi) atau 1 (terjadi). Persamaan di atas disebut juga sigmoid function.Model menggunakan kurva sigmoid berbentuk S untuk memetakan nilai input menjadi probabilitas.



Gambar 1. Kurva "S" dari Sigmoid Function

Sumber: https://www.ibm.com/think/topics/logistic-regression

Sigmoid function juga dapat dituliskan dalam bentuk berikut:

$$f\,(x, w) = \frac{1}{1 + e^{-\left(\sum_{i=0}(w_i\, x_i)\ \right)}}$$

dengan *w*i = βi yaitu weight atau koefisien tiap fitur dan *f* sesuai definisi P(x) di atas. Koefisien untuk *w*0 = β0 disebut juga bias.

Logistic regression menggunakan BCE (Binary Cross Entropy) untuk menghitung error dari prediksi model dan menyesuaikan parameternya, yaitu weight dan bias. Formula BCE adalah sebagai berikut.

$$BCE = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right]$$

dengan m adalah jumlah data (Number of Examples). Pada model ini, dapat diterapkan juga regularization, yaitu teknik untuk mengurangi pengaruh variabel yang kurang penting dengan cara mengecilkan koefisiennya. Beberapa teknik regularization adalah l1/Lasso dan l2/Ridge. Lasso (Least Absolute Shrinkage and Selection Operator) menambahkan nilai absolute dari besar koefisien sebagai penalty ke loss function. Sementara itu, Ridge menambahkan nilai kuadrat koefisien sebagai penalty ke loss function. Lasso dapat mengecilkan beberapa koefisien hingga nol yang dapat digunakan untuk memilih hanya fitur-fitur penting saja, sedangkan ridge dapat menghandle multikolinearitas dengan mengecilkan koefisien dari fitur-fitur yang saling berkorelasi, bukan menghapusnya. Dengan demikian, BCE dengan l1 dan l2 regularization secara berturut-turut adalah sebagai berikut.

$$BCE_{L1} = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] + \lambda\sum_{j=1}^{n}|w_j|$$

$$BCE_{L2} = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] + \frac{\lambda}{2}\sum_{j=1}^{n}w_j^2$$

dengan m adalah number of examples n adalah number of features.

Untuk menyesuaikan nilai parameter, dalam hal ini weight dan bias, dilakukan **gradient descent** atau **newton methods**. Pada gradient descent, parameter diperbarui secara iteratif dengan bergerak ke arah negatif dari gradien fungsi loss. Dengan prinsip **chain rules,** didapatkan bahwa untuk setiap iterasi dilakukan perhitungan berikut.

$$dw = \frac{1}{m}X^T(\hat{y}-y), \quad db = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)$$

Dengan melibatkan regularization, didapatkan:

$$dw \leftarrow dw + \lambda\,\mathrm{sign}(w)$$

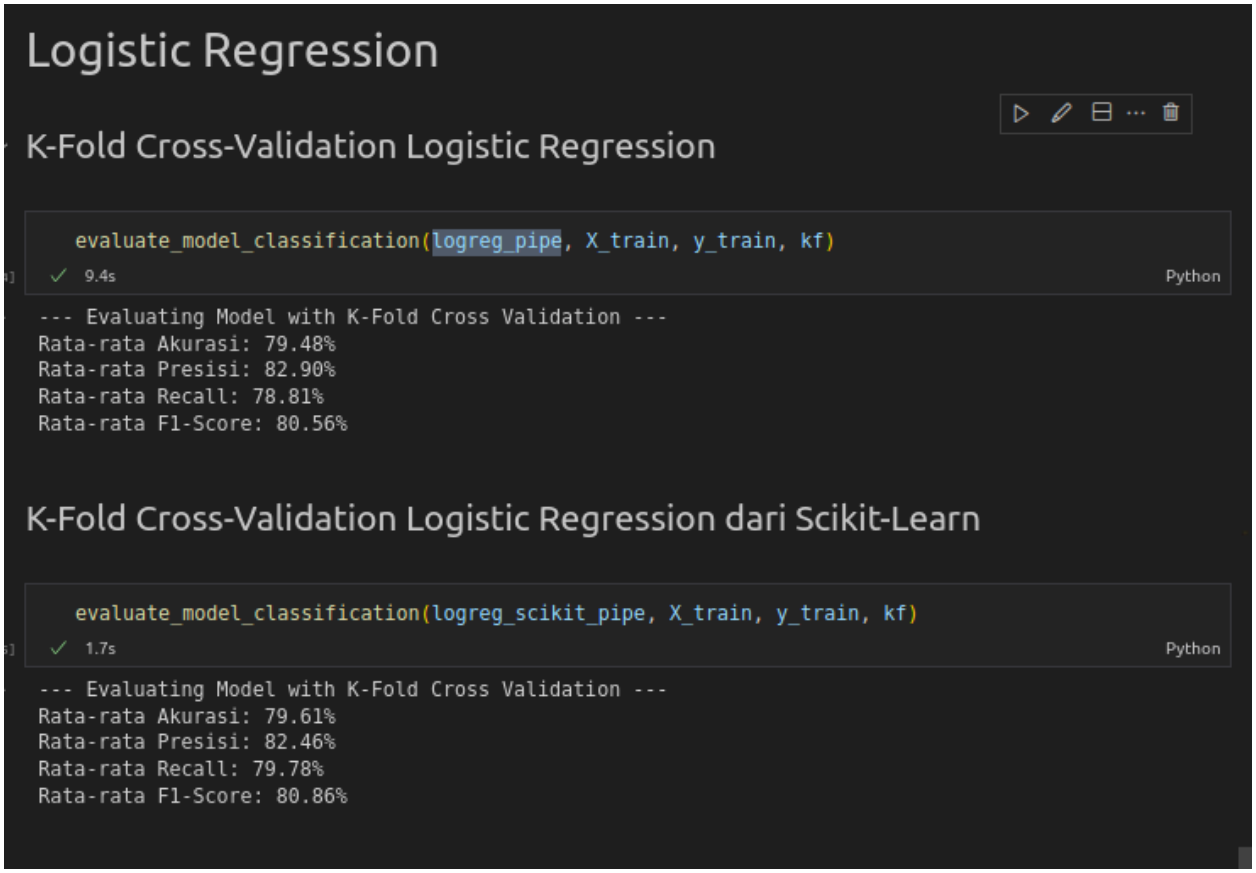
$$dw \leftarrow dw + \lambda w$$

sehingga parameter diperbarui dengan:

$$w \leftarrow w - \alpha\, dw, \quad b \leftarrow b - \alpha\, db$$

dengan α adalah **learning rate.**

Perbandingan hasil evaluasi model dari scratch dengan dari Scikit-Learn



Gambar 2. K-Fold Cross-Validation LogReg dari Scratch dan LogReg dari Sklearn

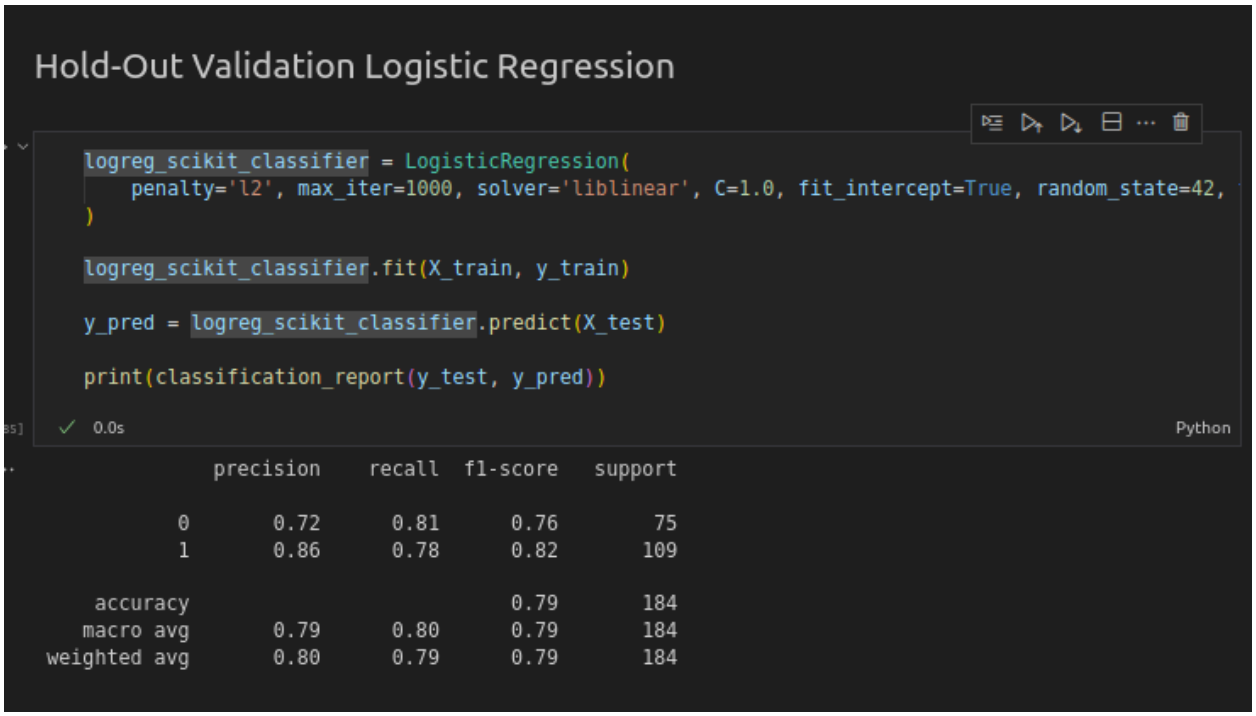
Sumber: Penulis

Skor evaluasi dari kedua model untuk 10-Fold Validation hampir sama di semua metrik. Hal ini membuktikan bahwa implementasi logistic regression dari scratch telah berhasil dan akurat,

karena mampu menghasilkan performa mirip Scikit-Learn.

Gambar 3. Hold-Out Validation LogReg dari Scratch

Sumber: Penulis



Gambar 4. Hold-Out Validation LogReg dari Sklearn

Sumber: Penulis

Dalam hold-out validation, model yang dibuat dari scratch menunjukkan performa yang lebih baik daripada model Scikit-learn. Perbedaan ini terjadi karena perbedaan implementasi seperti misalnya implementasi solvernya.

Ruang Improvement

- Optimasi solver selain gradient descent basic dan newton's method seperti adaptive learning rate.
- Memanfaatkan optimisasi komputasi secara penuh seperti library Numpy dan library perhitungan matematikanya.
- Masih mungkin untuk dicoba random state dan initialization.
- Masih bisa autotuning lambda dan alpha menggunakan grid search atau random search.

Referensi:

[1] *What Is Logistic Regression?*, IBM Think, 14 Mei 2025. [Daring]. Tersedia: https://www.ibm.com/think/topics/logistic-regression. [Diakses: 2 September 2025].

[2] *Understanding Logistic Regression*, Datagran oleh Necati Demir (Chief AI Officer). [Daring]. Tersedia: https://blog.datagran.io/posts/understanding-logistic-regression. [Diakses: 2 September 2025].

[3] *Regularization in Machine Learning*, GeeksforGeeks, diperbarui 2 Agustus 2025. [Daring]. Tersedia: https://www.geeksforgeeks.org/machine-learning/regularization-in-machine-learning/. [Diakses: 2 September 2025].

[4] *Logistic Regression From Scratch*, Medium oleh Koushik Ahmed Kushal, 28 Agustus 2023. [Daring]. Tersedia: https://medium.com/@koushikkushal95/logistic-regression-from-scratch-dfb8527a4226. [Diakses: 2 September 2025].