

1. Algoritma SoftMax Regression yang diterapkan merupakan generalisasi dari algoritma Logistic Regression untuk masalah multi class classification. Model ini bekerja dengan mempelajari hubungan linear antara fitur input dan probabilitas setiap kelas, kemudian menggunakan fungsi softmax untuk mengubah output linear menjadi distribusi probabilitas. Cara kerjanya adalah
 - a. Inisialisasi parameter
 - i. optimizer: Metode optimisasi ('gd' untuk gradient descent atau 'newton' untuk Newton's method)
 - ii. lr: Learning rate untuk mengontrol kecepatan pembelajaran
 - iii. n_iters: Jumlah iterasi training
 - iv. regularization: Jenis regularisasi (None atau 'l2')
 - v. lambda_param: Parameter kekuatan regularisasi
 - b. Training data (Fit method)
 - i. Mengidentifikasi unique classes dari data
 - ii. Membuat mapping dari label ke index numerik
 - iii. Mengconvert label ke one-hot encoding
 - iv. Memilih metode optimasi berdasarkan parameter
 - c. Dua Metode Optimasi
 - i. Gradient Descent
 1. Forward pass: pertama, model menghitung skor linear ($Z = XW + b$) dengan mengalikan fitur input dengan weight dan menambahkan bias. Kedua, skor linear ini diubah menjadi probabilitas menggunakan fungsi softmax yang mengeksponensialkan dan menormalkan skor. Ketiga, dihasilkan distribusi probabilitas dimana nilai tertinggi menentukan kelas prediksi.
 2. Hitung error: Selisih antara predicted vs actual probabilities
 3. Hitung gradien: Derivatif loss terhadap weights dan bias
 4. Tambahkan regularisasi: L2 regularization jika diaktifkan
 5. Update parameters: Adjust weights dan bias berdasarkan gradien
 - ii. Newton's Method
 1. Hitung gradien dan Hessian: Menggunakan second-order derivatives
 2. Per-class optimization: Optimisasi terpisah untuk setiap kelas

3. Matrix inversion: Invers Hessian matrix untuk convergence lebih cepat
4. Fallback ke GD: Jika Hessian singular (tidak bisa diinvers)

d. Prediction

- i. Hitung linear combination of features and weights
 - ii. Apply softmax untuk mendapatkan probabilities
 - iii. Pilih class dengan probability tertinggi
 - iv. Convert kembali ke label original
2. Berdasarkan hasil evaluasi, model dari Sklearn memiliki nilai Recall 0.64 dan F1 score 0.65, sementara untuk model yang saya buat sendiri mendapatkan nilai Recall 0.70 dan F1 score 0.71. Hal ini menunjukkan performa model yang saya buat lebih baik dikarenakan penggunaan optimizer yang berbeda, adanya penambahan regularisasi dan jumlah iterasi yang lebih banyak, sehingga model dapat memahami data dengan lebih baik.
3. Improvement yang dapat dilakukan
- a. Hyperparameter tuning dengan menggunakan tools seperti Grid Search, Optuna, dll untuk mendapatkan kombinasi parameter terbaik untuk model pada dataset ini.