REPOSITOR, Vol. 7, No. 2, Mei 2025, Pp. 169-176
ISSN : 2714-7975
E-ISSN : 2716-1382

Penerapan K-Nearest Neighbors Pada Single Layer Perceptron Untuk Klasifikasi Dataset Kecil Dengan Tiga Fitur

Arjuna Ranma Putra[*1], Devi Yunita [2], Julya Rahmah Shanty[3], Maria Iodiana Leki[4], Muhammad Aqshal Hidayat Fizhilillah[5], Rifky Firmansyah[6]
[1,2,3,4,5,6]Universitas Muhammadiyah Malang
arjunaputra1507@gmail.com[*]

Abstrak

Dalam pengembangan model machine learning untuk dataset kecil, pemilihan metode yang tepat menjadi kunci untuk menghasilkan klasifikasi yang akurat. Penelitian ini mengaplikasikan algoritma Single Layer Perceptron (SLP) untuk mengklasifikasikan dataset kecil dengan tiga fitur utama, yaitu Feature1, Feature2, dan Feature3. Algoritma SLP digunakan untuk mempelajari pola dalam data, dengan evaluasi model menggunakan teknik validasi silang k-fold. Teknik ini memastikan setiap bagian data digunakan sebagai data uji dan pelatihan secara bergantian, untuk mendapatkan hasil evaluasi yang lebih akurat. Selain itu, algoritma k-Nearest Neighbor (k-NN) digunakan untuk mencari nilai parameter K yang optimal guna meningkatkan akurasi model. Penelitian ini menggunakan 13 data sampel untuk melatih dan menguji model. Dengan pendekatan ini, diharapkan dapat meningkatkan kinerja model dalam mengklasifikasikan dataset kecil dengan tiga fitur.

Kata Kunci: *Single Layer Perceptron k-Nearest Neighbor Validasi Silang Klasifikasi Dataset kecil*

Abstract

In developing machine learning models for small datasets, choosing the right method is key to producing accurate classification. This research applies the Single Layer Perceptron (SLP) algorithm to classify a small dataset with three main features, namely Feature1, Feature2, and Feature3. The SLP algorithm is used to learn patterns in the data, with model evaluation using the k-fold cross-validation technique. This technique ensures each piece of data is used as test and training data in turn, to obtain more accurate evaluation results. In addition, the k-Nearest Neighbor (k-NN) algorithm was used to find the optimal K parameter value to improve the accuracy of the model. This study used 13 sample data to train and test the model. With this approach, it is expected to improve the performance of the model in classifying small datasets with three features.

Keywords: *Single Layer Perceptron k-Nearest Neighbor Cross Validation Small Dataset Classification*

1. Pendahuluan

Dalam bidang pembelajaran mesin, klassifikasi data seringkali menjadi tugas utama yang dihadapi, terutama ketika bekerja dengan dataset kecil yang memiliki fitur terbatas. Berbagai metode telah dikembangkan untuk mengatasi tantangan ini, salah satunya adalah K-Nearest Neighbors (K-NN) [1]. K-NN mengelompokkan data berdasarkan kedekatannya dalam ruang fitur, dan meskipun metode ini cukup efektif, kinerjanya sangat dipengaruhi oleh pemilihan parameter K serta karakteristik dari dataset yang digunakan.

Di sisi lain, perceptron satu lapis (Single Layer Perceptron - SLP) [2] merupakan model jaringan saraf sederhana yang sering digunakan untuk klasifikasi biner. Meskipun SLP efektif pada dataset sederhana, model ini cenderung kurang optimal ketika diterapkan pada dataset yang lebih kompleks atau memiliki jumlah fitur yang lebih banyak.

Seringkali, kombinasi antara metode K-NN dan SLP [3] dapat memberikan solusi yang lebih baik, terutama pada dataset kecil dengan jumlah fitur terbatas. Pendekatan ini memiliki potensi untuk meningkatkan akurasi klasifikasi dan memaksimalkan kinerja model.

2. Metode Penelitian

2.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini berjudul “Single Layer Perceptron” yang tersedia di Kaggle dan dataset ini dibuat oleh Abir Hasan.

Dataset ini terdiri dari lima record dengan atribut yang beragam, digunakan sebagai bahan utama dalam penelitian. Struktur dataset memiliki pola data dengan empat atribut utama:

1) Pattern
2) Feature1
3) Feature2
4) Feature3

Table 1. Struktur Data yang digunakan

Column	Tipe Data
Pattern	int
Feature 1	int
Feature 2	float
Feature 3	float
Class label	int

Pattern	Feature1	Feature2	Feature3	Class_Label
1	1	0.08	0.72	1
2	1	0.1	1	1
3	1	0.26	0.58	1
4	1	0.35	0.95	0
5	1	0.45	0.15	1
6	1	0.6	0.3	1
7	1	0.7	0.65	0
8	1	0.92	0.45	0
9	1	0.42	0.85	0
10	1	0.65	0.55	0
11	1	0.2	0.3	1
12	1	0.2	1	0
13	1	0.85	0.1	1

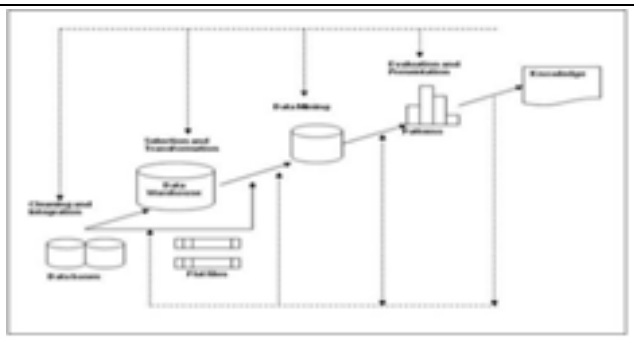
Gambar 1. Dataset yang digunakan

2.2 Preprocessing Data

Preprocessing data adalah langkah penting dalam mempersiapkan data mentah agar siap digunakan dalam penelitian. Tahapan ini bertujuan untuk memastikan data yang akan dianalisis dapat memberikan hasil yang optimal. Pada fase ini, dilakukan identifikasi terhadap nilai-nilai yang hilang dalam dataset. Nilai-nilai yang hilang kemudian ditangani melalui imputasi atau penghapusan, tergantung pada kebutuhan dan sifat dataset. Proses ini memastikan bahwa data yang digunakan bebas dari masalah yang dapat mengganggu akurasi analisis dan klasifikasi.

2.3 Data Mining

Data mining adalah tahap dalam Knowledge Discovery in Database (KDD) yang bertujuan mengekstraksi informasi berharga dari data. Dalam penelitian ini, tahapan KDD diterapkan untuk mengklasifikasikan dataset kecil dengan tiga fitur utama: Feature1, Feature2, dan Feature3[1].



Gambar 2. Tahapan KDD

1) ***Preprocessing*:** Data diperiksa untuk menghilangkan redundansi dan nilai kosong agar hasil analisis lebih akurat.
2) ***Transformation*:** Data diubah agar sesuai dengan format input yang dibutuhkan oleh algoritma K-NN dan SLP.
3) **Data Mining:** Algoritma K-NN dan SLP diterapkan untuk mengklasifikasikan data berdasarkan kedekatan dan pola yang ada.
4) ***Interpretation / Evaluation*:** Evaluasi dilakukan menggunakan teknik validasi silang untuk menilai akurasi model yang diterapkan.
5) ***Selection*:** Data yang digunakan terdiri dari 13 pola dengan tiga fitur utama dan label kelas. Data ini dipilih untuk memastikan relevansi dan kesesuaian dengan tujuan klasifikasi.

2.4 Klasifikasi

Analisis data, seperti klasifikasi, digunakan untuk memperkirakan label kelas yang tepat untuk sebuah sampel. Beberapa metode klasifikasi yang populer meliputi K-Nearest Neighbor, klasifikasi Bayesian, dan jaringan saraf. Teknik-teknik ini banyak diterapkan dalam pembelajaran mesin, sistem pakar, serta analisis statistik[1].

2.5 K-Nearest Neighbor (KNN)

Algoritma k-Nearest Neighbor (kNN) adalah metode klasifikasi berdasarkan jarak terdekat antara data uji dan data pelatihan. Langkah-langkahnya:

1) Menentukan Parameter K
   umlah tetangga terdekat dipilih berdasarkan pengujian nilai K ideal.
2) Menghitung Jarak Euclidean
   Menghitung jarak antara data uji dan data pelatihan menggunakan rumus:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

3) Mengurutkan Jarak Terdekat
   Data pelatihan diurutkan berdasarkan jarak terkecil ke data uji.
4) Menentukan Kategori Tetangga
   Mengambil kategori dari K tetangga terdekat.
5) Klasifikasi Mayoritas
   Data uji diklasifikasikan ke kelas dengan kategori mayoritas dari tetangga terdekat.

3. Hasil Penelitian dan Pembahasan

3.1 Tools yang Digunakan

Dalam penelitian ini, beberapa tools yang digunakan untuk implementasi dan analisis adalah sebagai berikut:

1) **Jupyter Notebook:** Digunakan untuk implementasi model Single Layer Perceptron (SLP), preprocessing data, dan perhitungan evaluasi akurasi.
2) **RapidMiner:** Digunakan untuk implementasi model K-Nearest Neighbors (K-NN), serta untuk membagi data menjadi data training dan data testing.
3) **Python Libraries:** Seperti numpy, matplotlib, dan pandas untuk melakukan manipulasi data dan visualisasi.

3.2 Preprocessing Data

Pada tahap ini, data yang digunakan untuk eksperimen ini dipersiapkan melalui proses preprocessing. Data yang digunakan terdiri dari tiga fitur yang berbeda, dan telah dilakukan normalisasi untuk memastikan semua fitur berada pada skala yang serupa.

3.3 Model Single Layer Perceptron (SLP)

Pada tahap ini, dilakukan pembangunan model Single Layer Perceptron (SLP) untuk melakukan klasifikasi biner menggunakan data yang telah diproses pada tahap sebelumnya. Model ini memanfaatkan tiga fitur input, yaitu Feature1, Feature2, dan Feature3, untuk memprediksi label kelas (Class_Label) yang bersifat biner (0 atau 1).

A. Deskripsi Model

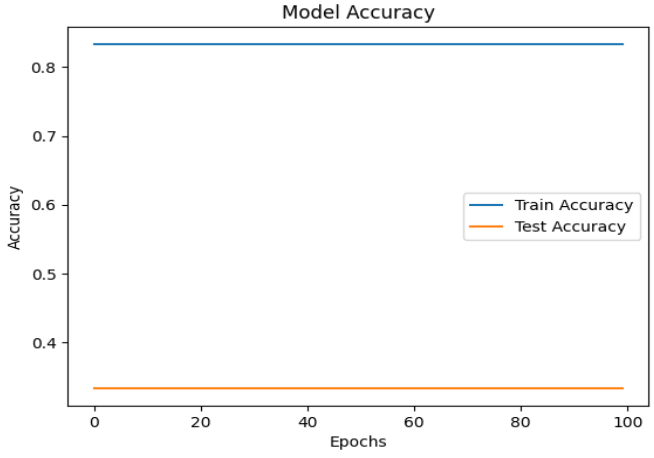
1) **Lapisan Input:** Menerima tiga fitur sebagai input.
2) **Lapisan Tersembunyi:** Terdiri dari 8 unit dengan fungsi aktivasi ReLU (Rectified Linear Unit), yang memungkinkan model untuk menangkap pola non-linear dalam data.
3) **Lapisan Output:** Menghasilkan output biner (0 atau 1) dengan menggunakan fungsi aktivasi sigmoid, yang sesuai untuk masalah klasifikasi biner.

B. Proses Pelatihan

Model dilatih menggunakan data pelatihan yang telah diproses dan dinormalisasi. Proses pelatihan berlangsung selama 100 epoch, dengan ukuran batch sebesar 1.

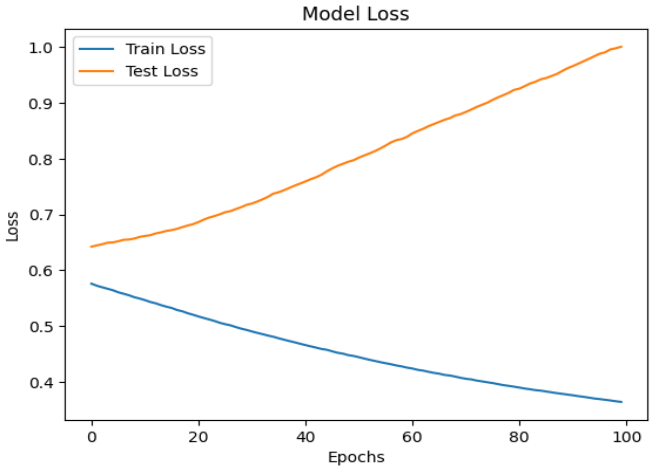
C. Hasil Evaluasi

Grafik Akurasi: Menampilkan perubahan akurasi model pada data pelatihan dan pengujian sepanjang epoch. Peningkatan akurasi menunjukkan bahwa model semakin baik dalam mengklasifikasikan data.



Gambar 3. Grafik Akurasi pada model

Grafik Loss: Menampilkan perubahan loss pada data pelatihan dan pengujian. Penurunan loss menunjukkan bahwa model semakin tepat dalam memprediksi hasil yang benar.



Gambar 4. Grafik loss pada Model

D. Pembagian Data

Data dibagi menggunakan metode train_test_split, yang membagi data menjadi 70% untuk pelatihan dan 30% untuk pengujian.

3.4 Penerapan Algoritma K-Nearest Neighbors pada Rapid Miner

Adversarial examples merupakan output setelah serangan adversarial attack dilakukan. Berikut merupakan tabel contoh adversarial examples dari setiap model yang diserang.

A. Data Training

Tabel 2. Data Training

Pattern	Feature1	Feature2	Feature3	Class_Label
1	1	0.8	0.72	1
2	1	0.1	1	1
3	1	0.26	0.58	1
4	1	0.35	0.95	0
5	1	0.45	0.15	1
6	1	0.6	0.3	1
7	1	0.7	0.65	0
8	1	0.92	0.45	0
9	1	0.42	0.85	0

B. Data Testing

Tabel 3. Data Testing

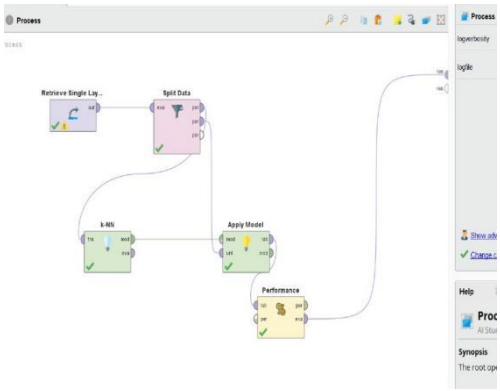
10	1	0.65	0.55	0
11	1	0.2	0.3	1
12	1	0.2	1	0
13	1	0.85	0.1	1

Langkah-Langkah:

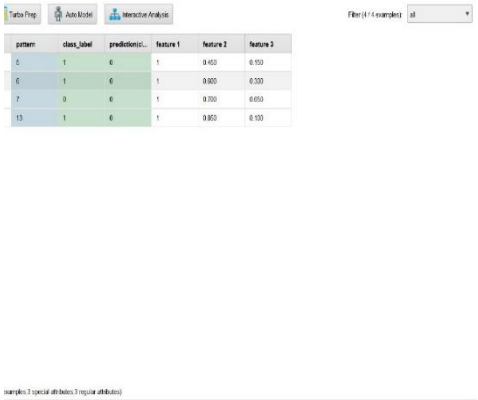
1) **Menentukan Parameter K:**
   Parameter K dipilih sebagai jumlah tetangga terdekat untuk klasifikasi data uji. Nilai K=9 dipilih untuk menghindari overfitting dan underfitting.
2) **Menghitung Jarak Euclidean:**
   Jarak antar data uji dan data latih dihitung menggunakan rumus Euclidean untuk menentukan kedekatan antara data.

3) **Menentukan Tetangga Terdekat:**
Data uji dibandingkan dengan data latih, dan K tetangga terdekat dipilih berdasarkan jarak terkecil.

4) **Klasifikasi Berdasarkan Mayoritas Kelas:**
Kelas data uji ditentukan berdasarkan mayoritas kelas dari K tetangga terdekat.



Gambar 5. Struktur Model

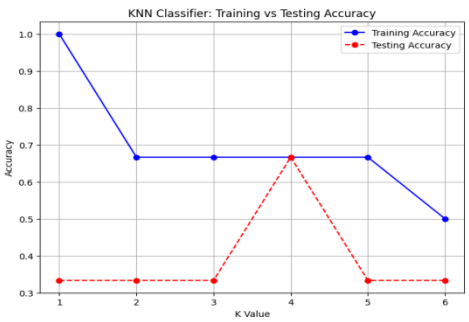


Gambar 6. Hasil Klasifikasi Data Uji Menggunakan Algoritma K-NN, dengan Kolom Prediction Mempermudah Identifikasi Prediksi Benar dan Salah

Gambar 5 dan Gambar 6 ini memvisualisasikan hasil klasifikasi model K-Nearest Neighbors (K-NN) pada data uji. Kolom Prediction (Cl.) menunjukkan hasil prediksi model yang dibandingkan dengan kolom Class_Label sebagai label asli. Format tabel ini mempermudah identifikasi baris dengan prediksi benar dan salah, serta mendukung analisis akurasi dan evaluasi performa model secara keseluruhan.

3.5 Hasil Evaluasi Model dan Perbandingan

Pada evaluasi ini, kami membandingkan akurasi model K-Nearest Neighbors (K-NN) pada data pelatihan dan pengujian dengan berbagai nilai K.



Gambar 7. Hasil Perbandingan Model

4. Kesimpulan

Berdasarkan eksperimen menggunakan algoritma K-Nearest Neighbors (K-NN) pada dataset terbatas, dapat disimpulkan hal-hal berikut:

1) **Overfitting pada K=1:** Akurasi pelatihan sempurna (1.0000), tetapi akurasi pengujian rendah (0.3333), menunjukkan overfitting, di mana model terlalu cocok dengan data pelatihan namun gagal menggeneralisasi.
2) **Underfitting pada K=2, K=3, dan K=5:** Model menunjukkan akurasi pelatihan dan pengujian rendah (0.6667 dan 0.3333), mengindikasikan model tidak cukup kompleks untuk menangkap pola dalam data.
3) **Keseimbangan pada K=4:** Model menunjukkan keseimbangan yang baik antara akurasi pelatihan dan pengujian (0.6667), menandakan kemampuan generalisasi yang baik.
4) **Penurunan pada K=6:** Akurasi pelatihan turun (0.5000), dengan akurasi pengujian tetap rendah (0.3333), menunjukkan model kehilangan kemampuan mempelajari pola data.

5. Saran

Diperlukan pengujian model pada dataset yang lebih besar dan kompleks untuk memahami kemampuannya dalam menangani data variatif. Penambahan fitur relevan dan eksplorasi teknik ekstraksi fitur dapat meningkatkan akurasi. Perbandingan dengan algoritma lain, seperti Decision Trees atau SVM, serta penerapan tuning hyperparameter dan regularisasi, penting untuk meningkatkan performa dan menghindari overfitting. Pengujian pada kasus dunia nyata, seperti data sensor atau medis, juga disarankan untuk memahami aplikasi praktis model ini.

Referensi

[1] Purwandi, R. (2023). *Penerapan Algoritma k-Nearest Neighbour (kNN) dalam Memprediksi Kelulusan Mahasiswa pada Konteks Praktikum di Laboratorium*. *Pros. Seminar Kecerdasan Artifisial, Sains Data, dan Pendidikan Masa Depan (PROKASDADIK)*, 1, 347. E-ISSN: 3063-5845.

[2] Iffah'da, A. N., & Desiani, A. (2022). *Implementasi Algoritma K-Nearest Neighbor (K-NN) dan Single Layer Perceptron (SLP) dalam Prediksi Penyakit Sirosis Biliary Primer*. *Jurnal Ilmiah Informatika (JIMI)*, 7(1), 65-74. https://doi.org/10.35316/jimi.v7i1.65-74. P-ISSN: 2549-7480, E-ISSN: 2549-6301.

[3] Riansa, D. A. (2016). *Pengenalan tanda tangan menggunakan algoritma single layer perceptron* (Sarjana thesis, Universitas Negeri Jakarta).