Implementasi Algoritma K-Nearest Neighbors dalam Klasifikasi Prediksi Diabetes

Shafa Auliya

Program Studi Ilmu Komputer, Fakultas Matematikan dan Ilmu Pengetahuan Alam, Universitas Lampung

Email : shafa.auliya21@students.unila.ac.id

Abstrak

Penelitian ini mengimplementasi algoritma K-Nearest Neighbors (KNN) dalam klasifikasi untuk memprediksi kemungkinan diabetes berdasarkan dataset yang diambil dari kaggle. Penelitian melibatkan pemisahan dataset menjadi data latih dan uji, diikuti dengan standarisasi fitur menggunakan metode StandardScaler. Evaluasi dilakukan melalui pengukuran akurasi, matriks konfusi, dan laporan klasifikasi. Temuan ini memberikan wawasan tentang potensi penggunaan KNN dalam prediksi diabetes dan memberikan pemahaman lebih lanjut tentang performa model pada konteks ini. Hasil penelitian ini dapat memberikan kontribusi pada pengembangan sistem prediksi penyakit dan mendukung pengambilan keputusan di bidang kesehatan.

Kata Kunci : K-Nearest Neighbors, Diabetes, Klasifikasi, Prediksi.

1. Pendahuluan

Diabetes adalah salah satu penyakit mematikan dan penyakit kronis dengan ditandai dengan peningkatan gula darah, penyakit ini semakin meluas tersebar di seluruh dunia dan memiliki dampak kesehatan serius yang berdampak buruk pada manusia (Karo & Hendriyana 2022). Penyakit diabetes ini disebabkan oleh ketidakmampuan tubuh untuk memproduksi

cukup insulin atau ketidakmampuan jaringan tubuh untuk menggunakan insulin dengan efektif, kondisi ini dapat mengakibatkan sejumlah masalah kesehatan yang serius termasuk risiko kematian dini (Ridwan & Setiawan 2023). Salah satu tanda utama dari penyakit diabetes ini adalah peningkatan kadar glukosa darah atau disebut *hiperglikemia*. Kadar glukosa dalam darah dapat bervariasi sepanjang hari, jika setelah makan kadar glukosa akan meningkat dan akan kembali normal setelah dua jam. Penderita diabetes mencapai 1.9% dari total populasi manusia di dunia (Triyono, dkk 2021). Setiap tahun pasien diabetes mengalami peningkatan yang cukup signifikan, di Indonesia kasus penyakit diabetes berada di peringkat ke-7 dalam hal prevalensi penyakit diabetes (Wibowo & Rahmawati 2023).

Banyaknya individu yang menderita diabet baru mendapatkan diagnosis setelah terjadi komplikasi. Dengan hal ini, menjadi salah satu faktor yang menyebabkan peningkatan jumlah penderita diabetes adalah keterlambatan dalam mendiagnosis penyakit diabetes. Oleh karena itu, diperlukan penggunaan teknologi *machine learning* untuk melakukan prediksi terhadap penyakit diabetes. Beberapa penelitian terdahulu seperti yang dilakukan oleh Putry & Sari mendapatkan akurasi sebesar 75% untuk klasifikasi diabetes menggunakan metode KNN. Selanjutnya ada penelitian dari Lestari, dkk yang juga melakukan penelitian dengan menggunakan dataset diabetes menggunakan metode KNN dengan nilai k = 23 mendapatkan akurasi sebesar 96%. Berdasarkan penjelasan dari penelitian terdahulu, penelitian ini akan menggunakan algoritma K-Nearest Neighbors (KNN) untuk mengklasifikasi deteksi diabetes dengan menggunakan dataset yang diambil dari Kaggle. Pemilihan KNN sebagai metode klasifikasi dilatarbelakangi oleh kecocokannya untuk penanganan dataset, sehingga diharapkan dapat memberikaan hasil yang optimal pada analisis deteksi diabetes pada dataset tersebut.

2. Metodologi

Metode yang diterapkan dalam penelitian ini dimulai dari tahap pengumpulan dataset, dilanjutkan dengan proses preprocessing data, penerapan metode K-Nearest Neighbors

(KNN), dan yang terakhir adalah evaluasi kinerja model sesuai dengan Langkah-langkah yang telah ditetapkan.

a. Pengumpulan Dataset

Dataset diabetes ini berasal dari National Institute of Diabetes and Digestive and Kidney yang bersumber dari *website Kaggle,* pada dataset diabetes memiliki 9 kolom dengan 768 data. Kolom ini terdiri dari Pregnancies, Glukose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome.

Features	Deskripsi	Kriteria
Pregnancies	Jumlah kehamilan	0 - 17
Glukose	Kadar glukosa dalam darah	0 - 199
BloodPressure	Pengukuran tekanan darah	0 - 122
SkinThickness	Ketebalan kulit	0 - 99
Insulin	Kadar insulin dalam darah	0 - 846
BMI	Indeks massa tubuh	0 – 67.1
DiabetesPedigreeFunctio	Persentase diabetes	0,08 – 2.42
Age	Usia pasien	21 – 81 tahun
Outcome	Hasil akhir 1 adalah Ya dan 0 adalah Tidak	0 dan 1

b. Tahap Pre-processing

Pada tahap ini dilakukan preprocessing dengan cara melakukan hapus kolom SkinThickness dan Outcome. Dari kolom yang sebelumnya berjumlah 10 menjadi 8 kolom.

```
X_train = df.drop(['Outcome', 'SkinThickness'], axis=1)
y_train = df['Outcome']
```

```
X_train
```

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	0	33.6	0.627	50
1	1	85	66	0	26.6	0.351	31
2	8	183	64	0	23.3	0.672	32
3	1	89	66	94	28.1	0.167	21
4	0	137	40	168	43.1	2.288	33
...	...	...	...	...	...	...	...
763	10	101	76	180	32.9	0.171	63
764	2	122	70	0	36.8	0.340	27
765	5	121	72	112	26.2	0.245	30
766	1	126	60	0	30.1	0.349	47
767	1	93	70	0	30.4	0.315	23

768 rows × 7 columns

c. Metode K-Nearest Neighbors (KNN)

Algoritma K-Nearest Neighbor merupakan salah satu metode klasifikasi data mining, KNN mengklasifikasikan sekumpulan data berdasarkan data pembelajaran diberi label.

Berikut langkah-langkan melakukan klasifikasi dengan menggunakan algoritma K-Nearest Neighbor :

1. Menentukan X_train dan y_train

```
X_train = df.drop(['Outcome', 'SkinThickness'], axis=1)
y_train = df['Outcome']
```

- X_train mewakili dataset fitur yang digunakan untuk melatih model. Setiap baris dari X_train mengandung nilai-nilai fitur yang diperlukan untuk membuat prediksi.

- y_train adalah variabel target (variabel dependen) yang sesuai dengan setiap baris dalam X_train. Ini berisi label atau nilai yang ingin diprediksi oleh model. Pada dasarnya, y_train menyimpan informasi target yang sesuai dengan setiap baris dalam X_train.

2. Membagi data menggunakan train_test_split dengan test size sebesar 0.2. Data yang digunakan untuk test adalah 20% dan train 80%

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.2, random_state=0)
```

3. Mencari nilai k terbaik

```
from sklearn.neighbors import KNeighborsClassifier
metrics = []

for k in range(1, 11):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    accuracy = knn.score(X_test, y_test)
    metrics.append(accuracy)

best_k = metrics.index(max(metrics)) + 1
best_k
```

```
7
```

4. Membangun model KNN menggunakan KNeighborsClassifier
   - Menentukan jumlah tetangga (k) yang akan digunakan dalam penentuan kelas, berdasarkan nilai terbaik (best_k) yang telah diperoleh sebelumnya.
   - Menghitung jarak antar data terhadap semua data training menggunakan jarak Euclidean
   - Mengurutkan hasil perhitungan
   - Menentukan tetangga terdekat berdasarkan jarak minimum ke k
   - Menentukan kategori dari tetangga terdekat dengan data

d. Evaluasi

Evaluasi dilakukan untuk melihat precision, recall, f1-score, support dengan menggunakan classification_report.

```
from sklearn.metrics import classification_report

print(classification_report(y_test, knn.predict(X_test)))
pd.crosstab(y_test, y_pred, rownames=['Actual'], colnames=['Predicted'], margins=True)
```

```
              precision    recall  f1-score   support

           0       0.81      0.89      0.85       107
           1       0.68      0.53      0.60        47

    accuracy                           0.78       154
   macro avg       0.74      0.71      0.72       154
weighted avg       0.77      0.78      0.77       154
```

Predicted	0	1	All
Actual			
0	94	13	107
1	16	31	47
All	110	44	154

3. Hasil dan Pembahasan

Jumlah K	Hasil Akurasi
1	0.72%
2	0.77%
3	0.73%
4	0.75%
5	0.76%
6	0.80%
7	0.81%
8	0.81%
9	0.80%

10	0.77%

Dapat dilihat dari tabel diatas yang memiliki akurasi tertinggi adalah jumlah nilai k=7 dan k=8 dan pada best k yang sudah dilakukan dengan metrics adalah nilai k=7.

4. Kesimpulan

Dalam implementasi algoritma K-Nearest Neighbors (KNN) untuk klasifikasi prediksi diabetes, penelitian ini telah menunjukkan langkah-langkah yang komprehensif dalam membangun dan mengevaluasi model. Pada awalnya, dataset ini diambil dari website kaggle, diikuti dengan pemisahan fitur dan variabel target. Proses preprocessing dilakukan untuk meningkatkan kualitas data, termasuk standardisasi fitur menggunakan metode StandardScaler. Model KNN kemudian dibangun dengan variasi jumlah tetangga, dan evaluasi dilakukan dengan mengukur akurasi, matriks konfusi, dan laporan klasifikasi.

Hasil evaluasi membantu menentukan jumlah tetangga terbaik (best_k) yang memberikan performa model optimal. Penggunaan best_k selanjutnya diaplikasikan pada model KNN untuk melakukan prediksi pada dataset uji. Hasilnya memberikan pemahaman yang lebih mendalam terkait kemampuan model dalam mengklasifikasikan apakah seseorang berisiko terkena diabetes atau tidak. Dengan hal ini, untuk dataset ini mendapatkan nilai akurasi sebesar 0.81% dengan nilai k=7.

Daftar Pustaka

Karo Karo, I. M., & Hendriyana, H. (2022). Klasifikasi Penderita diabetes Menggunakan ALGORITMA machine learning Dan Z-Score. *Jurnal Teknologi Terpadu*, *8*(2), 94–99. https://doi.org/10.54914/jtt.v8i2.564

Ridwan, A. M., & Setyawan, G. D. (2023a). Perbandingan berbagai model machine learning Untuk Mendeteksi diabetes. *TEKNOKOM*, *6*(2), 127–132. https://doi.org/10.31943/teknokom.v6i2.152

Triyono, A., Trianto, R. B., & Arum, D. M. P. (2021). EARLY DETECTION OF DIABETES MELLITUS USING RANDOM FOREST ALGORITHM. *Julia : Jurnal Ilmu Komputer An Nuur*, *1*(01), 25–31.

Wibowo, A., & Rahmawati, S. (2023). Evaluasi Model Klasifikasi Algoritma Terbimbing Kuantitatif terhadap Penyakit Diabetes. *Journal of Information Technology and Computer Science*, *8*(3), 127–134.