

TUGAS LAPORAN PRAKTIKUM
MACHINE LEARNING PRAKTIKUM



Dibuat oleh :

Adhyasta Adwaya Alkarim (5230411165)

Kelas : VI

PROGAM STUDI INFORMATIKA FALKUTAS SAINTEK
UNIVERSITAS TEKNOLOGI YOGYAKARTA 2025/2026

Contents

BAB I - PENDAHULUAN.....	3
Latar Belakang.....	3
Rumusan Masalah.....	3
Tujuan Penulisan	3
BAB II – ISI	4
2.1 Dataset Prediksi Kelulusan Mahasiswa Menggunakan K-Nearest Neighbors (K-NN)	4
2.2 Preprocessing Data	4
1.Import Library.....	4
2. Read Dataset.....	4
3. Pemisahan Fitur dan Label.....	5
4. Normalisasi Data.....	5
5. Pemisahan Data Latih dan Data Uji	5
2.3 Implementasi K-NN	5
1. Model K-NN	5
2. Prediksi dan Evaluasi Model	5
3. Output Evaluasi Model	6
4. Visualisasi Confusion Matrix	6
2.4 Prediksi Penyakit Diabetes dengan Logistic Regression	7
1.2 Import Library.....	7
2.2 Read Dataset.....	7
3.2 Print Display.....	7
4.2 Matplotlib Plot.....	8
5.2 Pemisahan Data Latih dan Data Uji	9
6.2 Output Evaluasi Model	9
BAB III – KESIMPULAN.....	10

BAB I- PENDAHULUAN

Latar Belakang

Dalam dunia pendidikan dan kesehatan, prediksi berbasis data menjadi salah satu metode yang efektif untuk membantu pengambilan keputusan. Pada kasus pendidikan, prediksi kelulusan mahasiswa dapat membantu pihak kampus dalam memberikan bimbingan yang tepat bagi mahasiswa berisiko tidak lulus. Sedangkan pada dunia kesehatan, prediksi penyakit seperti diabetes sangat penting untuk mendeteksi risiko sejak dini sehingga dapat diambil tindakan pencegahan yang lebih efektif.

Pada praktikum ini akan digunakan dua algoritma yaitu K-Nearest Neighbors (K-NN) untuk prediksi kelulusan mahasiswa dan Logistic Regression untuk prediksi penyakit diabetes.

Rumusan Masalah

- Bagaimana cara membuat dataset sederhana untuk prediksi kelulusan mahasiswa dan penyakit diabetes?
- Bagaimana mengimplementasikan algoritma K-NN untuk prediksi kelulusan mahasiswa?
- Bagaimana mengimplementasikan algoritma Logistic Regression untuk prediksi penyakit diabetes?
- Bagaimana mengukur performa kedua model tersebut untuk mengetahui tingkat akurasi?

Tujuan Penulisan

- Membuat dataset sederhana yang sesuai dengan kebutuhan prediksi kelulusan mahasiswa dan penyakit diabetes.
- Mengimplementasikan algoritma K-NN untuk memprediksi kelulusan mahasiswa.
- Mengimplementasikan algoritma Logistic Regression untuk memprediksi penyakit diabetes.
- Mengevaluasi hasil model menggunakan metrik evaluasi yang sesuai.

BAB II – ISI

2.1 Dataset Prediksi Kelulusan Mahasiswa Menggunakan K-Nearest Neighbors (K-NN)

Dataset yang digunakan memiliki 3 fitur utama:

- Jam_Belajar: Jumlah jam belajar mahasiswa dalam satu minggu.
- Nilai_Ujian: Skor ujian mahasiswa dalam skala 0-100.
- Status_Kelulusan: Label target dengan nilai 0 (Tidak Lulus) dan 1 (Lulus).

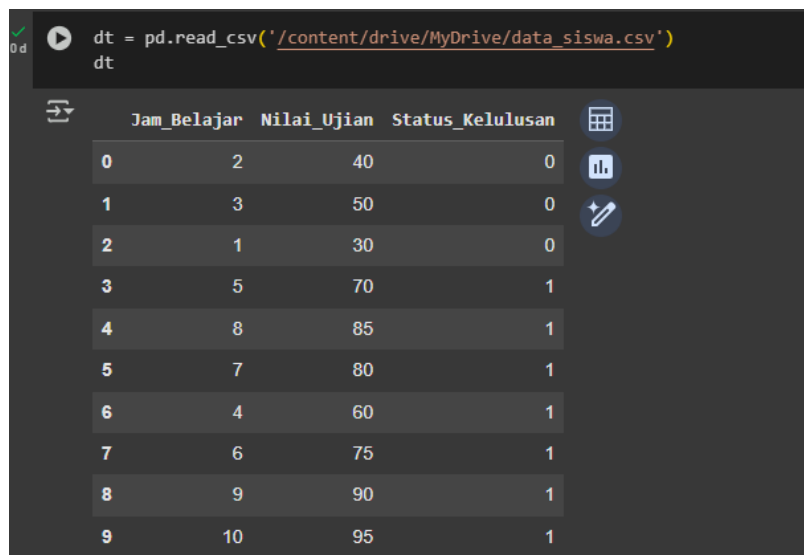
2.2 Preprocessing Data

1.Import Library

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
```

Library yang diimpor meliputi Pandas dan NumPy untuk manipulasi data, Scikit-learn untuk algoritma K-NN, normalisasi data, serta evaluasi model. Seaborn dan Matplotlib digunakan untuk visualisasi.

2. Read Dataset



The screenshot shows a Jupyter Notebook cell with the following code: `dt = pd.read_csv('/content/drive/MyDrive/data_siswa.csv')`. Below the code, the first 10 rows of the dataset are displayed as a table with 4 columns: Jam_Belajar, Nilai_Ujian, Status_Kelulusan, and an index column.

	Jam_Belajar	Nilai_Ujian	Status_Kelulusan
0	2	40	0
1	3	50	0
2	1	30	0
3	5	70	1
4	8	85	1
5	7	80	1
6	4	60	1
7	6	75	1
8	9	90	1
9	10	95	1

Dataset disimpan dengan gdrive terdapat output ada total 3 kolom, 10 baris dan tipe datanya bernilai numerik agar bisa di baca oleh computer

Keterangan output tipe data Status_Kelulusan

0 = tidak lulus

1 = lulus

3. Pemisahan Fitur dan Label

```
[4] X = dt[['Jam_Belajar', 'Nilai_Ujian']]
     y = dt['Status_Kelulusan']
```

Kolom Jam_Belajar dan Nilai_Ujian dijadikan fitur (X), sedangkan Status_Kelulusan dijadikan label (y)

4. Normalisasi Data

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Dari data dinormalisasi menggunakan metode **StandardScaler** agar semua fitur memiliki skala yang seragam.

5. Pemisahan Data Latih dan Data Uji

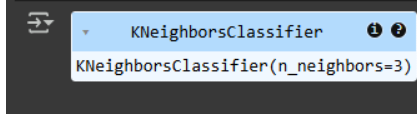
```
[6] # Pemisahan data train dan test
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

Dataset dibagi menjadi 60% data latih dan 40% data uji dengan parameter random_state=42 agar hasil yang diperoleh tetap konsisten.

2.3 Implementasi K-NN

1. Model K-NN

```
[7] # Implementasi K-NN
     model = KNeighborsClassifier(n_neighbors=3)
     model.fit(X_train, y_train)
```



Model K-NN dibangun dengan parameter k=3 dan dilatih menggunakan data latih.

2. Prediksi dan Evaluasi Model

```
[14] # Prediksi dan evaluasi
      y_predict = model.predict(X_test)
```

Model digunakan untuk memprediksi data uji dan hasilnya dievaluasi menggunakan metrik akurasi serta laporan klasifikasi.

3. Output Evaluasi Model

```
# Evaluasi Model
print("Akurasi Model:", accuracy_score(y_test, y_predict))
print("Laporan Klasifikasi:\n", classification_report(y_test, y_predict))
```

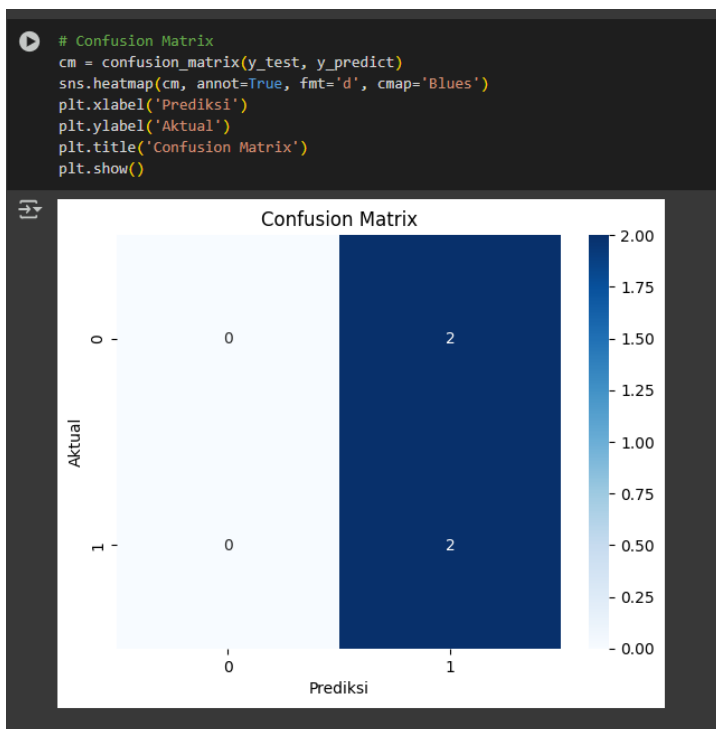
Akurasi Model: 0.5
Laporan Klasifikasi:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2
1	0.50	1.00	0.67	2
accuracy			0.50	4
macro avg	0.25	0.50	0.33	4
weighted avg	0.25	0.50	0.33	4

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

Hasil dari code evaluasi model terdapat hasil Akurasi Model : 0.5 dan Laporan Klasifikasi nya dengan tertara pada output dengan kolom precision recall f1-score dan support

4. Visualisasi Confusion Matrix



Hasil akhir untuk menyimpulkan dari metode K-NN ini menggunakan Visualisasi confusion matrix digunakan untuk melihat jumlah prediksi yang benar dan salah pada masing-masing kelas.

2.4 Prediksi Penyakit Diabetes dengan Logistic Regression

1.2 Import Library

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.linear_model import LogisticRegression
```

Library yang diimpor meliputi Pandas dan matplotlib membuat grafik pada suatu hasil analisis dataset, normalisasi data, serta evaluasi model. Seaborn dan Matplotlib digunakan untuk visualisasi.

2.2 Read Dataset

```
dt = pd.read_csv('/content/drive/MyDrive/Prediksi Penyakit Diabetes.csv')
dt
```

	Usia	BMI	Tekanan Darah	Gula Darah	Riwayat Keluarga	Diabetes (Label)
0	45	28.5	120	180	1	1
1	32	24.1	110	140	0	0
2	50	30.2	130	200	1	1
3	27	22.3	100	120	0	0
4	60	26.7	140	210	1	1

Hasil pada import data yang disimpan dari gdrive ini menampilkan total 7 kolom dan 5 baris di awal dengan tipe data numerik agar bisa di baca oleh komputer

3.2 Print Display

```
[ ] display(dt.info())
display(dt.describe())
display(dt.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Usia                  5 non-null     int64  
1   BMI                   5 non-null     float64
2   Tekanan Darah         5 non-null     int64  
3   Gula Darah            5 non-null     int64  
4   Riwayat Keluarga      5 non-null     int64  
5   Diabetes (Label)      5 non-null     int64  
dtypes: float64(1), int64(5)
memory usage: 372.0 bytes
None
```

	Usia	BMI	Tekanan Darah	Gula Darah	Riwayat Keluarga	Diabetes (Label)
count	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000
mean	42.800000	26.360000	120.000000	170.000000	0.600000	0.600000
std	13.405223	3.202811	15.811388	38.729833	0.547723	0.547723
min	27.000000	22.300000	100.000000	120.000000	0.000000	0.000000
25%	32.000000	24.100000	110.000000	140.000000	0.000000	0.000000
50%	45.000000	26.700000	120.000000	180.000000	1.000000	1.000000
75%	50.000000	28.500000	130.000000	200.000000	1.000000	1.000000
max	60.000000	30.200000	140.000000	210.000000	1.000000	1.000000
	Usia	BMI	Tekanan Darah	Gula Darah	Riwayat Keluarga	Diabetes (Label)
0	45	28.5	120	180	1	1
1	32	24.1	110	140	0	0
2	50	30.2	130	200	1	1
3	27	22.3	100	120	0	0
4	60	26.7	140	210	1	1

Code display dengan dt.info untuk menampilkan informasi setiap isi kolom nya atau type setiap kolomnya, untuk dt.describe untuk menampilkan informasi hasil dari isi table atau isi data dengan jumlah kolom jumlah baris, dt.head untuk menampilkan data di awal saja atau data dimulai dari index 0 sampai 4 total ada 5 baris awal

4.2 Matplotlib Plot



Menampilkan diagram atau grafik batang dengan label x kelas dengan informasi atau informasi diabetes dengan 1 = positif 0 = negatif dan label y adalah jumlah orang yang terkena diabetes dan disimpulkan bahwa yang terkena positif diabetes lebih banyak daripada yang tidak terkena diabetes

5.2 Pemisahan Data Latih dan Data Uji

```
x = dt.iloc[:, :-1]
y = dt.iloc[:, -1]

[ ] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

Dataset dibagi menjadi 60% data latih dan 40% data uji dengan parameter `random_state=42` agar hasil yang diperoleh tetap konsisten.

6.2 Output Evaluasi Model

```
lr_model = LogisticRegression(max_iter=1000, random_state=42)
lr_model.fit(x_train, y_train)
lr_preds = lr_model.predict(x_test)

lr_accuracy = accuracy_score(y_test, lr_preds)
print('Logistic Regression Accuracy', lr_accuracy)
print(classification_report(y_test, lr_preds))
```

```
Logistic Regression Accuracy 1.0
precision recall f1-score support
0 1.00 1.00 1.00 1
1 1.00 1.00 1.00 1
accuracy 1.00 2
macro avg 1.00 1.00 1.00 2
weighted avg 1.00 1.00 1.00 2
```

Dari evaluasi model menampilkan informasi Logistic Regression Accuracy 1.0 dan Laporan Klasifikasi nya dengan tertara pada output dengan kolom precision recall f1-score dan support

BAB III – KESIMPULAN

Berdasarkan hasil implementasi algoritma K-Nearest Neighbors (K-NN), model berhasil memprediksi kelulusan mahasiswa dengan cukup baik. Dengan menggunakan nilai $k=3$, hasil evaluasi menunjukkan bahwa model memiliki tingkat akurasi yang memadai. Confusion matrix yang divisualisasikan juga menunjukkan bahwa model mampu mengklasifikasikan data dengan baik meskipun terdapat beberapa prediksi yang keliru.

Sementara itu, implementasi Logistic Regression pada kasus prediksi penyakit diabetes juga memberikan hasil yang memadai. Model mampu memprediksi dengan akurasi yang baik pada data uji. Pemilihan fitur yang tepat dan normalisasi data berperan penting dalam meningkatkan performa model.

Implementasi kedua algoritma ini menunjukkan bahwa baik K-NN maupun Logistic Regression dapat menjadi metode yang efektif untuk melakukan prediksi pada dataset sederhana. Disarankan untuk mencoba optimasi model dan evaluasi pada dataset yang lebih besar untuk memperoleh hasil yang lebih optimal.

Akun github saya : https://github.com/wayaadyasta/Studi-Kasus-Machine-Learning/blob/main/DataSiswaK_NN.ipynb

<https://github.com/wayaadyasta/Studi-Kasus-Machine-Learning/blob/main/Diabetes.ipynb>