

BAB V

PENUTUP

5.1 Kesimpulan

Dari penelitian ini didapatkan hasil bahwa SVM lebih baik dari KNN maka dapat disimpulkan beberapa point sebagai berikut :

1. Perbandingan Performa Algoritma KNN dan SVM: SVM lebih unggul dibandingkan KNN dalam memprediksi obesitas pada anggota Destroyer Muscle Gym. SVM menunjukkan performa yang sangat baik dengan metrik berikut akurasi testing 0.925, precision 0.926, recall 0.925, dan F1 score 0.924.
2. Faktor-faktor yang Mempengaruhi Akurasi Prediksi: Fitur seperti usia, berat badan, dan tinggi badan harus terdistribusi dengan baik untuk memprediksi obesitas secara akurat.
3. Efektivitas dan *Feedback* Pengguna Terhadap Model Prediksi Obesitas: SVM terbukti lebih efektif dan efisien, memudahkan staf dan pelatih, serta disukai anggota gym karena kemudahan penggunaan dan akurasinya.

5.2 Saran

Berdasarkan hasil penelitian ini, berikut adalah beberapa saran yang dapat diberikan:

1. Edukasi dan Pelatihan Staf: Melatih staf gym mengenai penggunaan dan interpretasi hasil model prediksi obesitas dapat meningkatkan efektivitas penggunaan model ini. Staf yang terlatih dapat memberikan panduan yang lebih baik kepada anggota gym berdasarkan hasil prediksi.

2. Pemeliharaan dan Pembaruan Model, meskipun model SVM telah menunjukkan hasil yang baik, penting untuk terus memantau performa model dan melakukan pembaruan secara berkala.

DAFTAR PUSTAKA

- Abdusyukur, F. (2023). PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) UNTUK KLASIFIKASI PENCEMARAN NAMA BAIK DI MEDIA SOSIAL TWITTER. *KOMPUTA: Jurnal Ilmiah Komputer Dan Informatika*, 12(1).
- Agustina, W., Lestari, R. M., & Prasida, D. W. (2023). Hubungan Aktivitas Fisik dengan Kejadian Obesitas pada Usia Produktif di Wilayah Kerja Puskesmas Marina Permai Kota Palangka Raya. *Jurnal Surya Medika*, 9(1), 1–8. <https://doi.org/10.33084/jsm.v9i1.5125>
- Azizah, S. N., & Yuliawati, R. (n.d.). Literature Review : Hubungan Antara Kebiasaan Olahraga dengan Kejadian Obesitas Pada Anak Sekolah. In *Borneo Student Research* (Vol. 3, Issue 3).
- Budianto, A., Ariyuana, R., & Maryono, D. (2019). PERBANDINGAN K-NEAREST NEIGHBOR (KNN) DAN SUPPORT VECTOR MACHINE (SVM) DALAM PENGENALAN KARAKTER PLAT KENDARAAN BERMOTOR. *Jurnal Ilmiah Pendidikan Teknik Dan Kejuruan*, 11(1), 27. <https://doi.org/10.20961/jiptek.v11i1.18018>
- Dinda Qatrunnada, R. (n.d.). *Faktor Penyebab Kejadian Kelebihan Berat Badan dan Obesitas pada Anak-anak dan Dewasa Factors that Cause Overweight and Obesity in School-Age Children and Adult*.
- Fuansah, Y. S., Meileni, H., Novianti, L., Manajemen Informatika, J., Sriwijaya, N., Srijaya, J., Bukit, N., & Palembang, B. (n.d.). *Yohana Sicke Fuansah, dkk., Implementasi Metode K-Nearest Neighbor ... 929 Implementasi Metode K-Nearest Neighbor untuk Menentukan Klasifikasi Status Ekonomi Penerima Bantuan*.
- Ida Niara, S., Pertiwi Program Studi Kesehatan Masyarakat, Y., Ilmu Kesehatan, F., & Pembangunan Nasional Veteran Jakarta Abstrak, U. (n.d.). Pencegahan Obesitas pada Remaja Melalui Intervensi Promosi Kesehatan: Studi Literatur Prevention of

Obesity in Adolescents Through Health Promotion Interventions: Literature Study. In *Jurnal Ilmiah Kesehatan Masyarakat* (Vol. 14).

Oroh, P. J., Wungow, H. I. S., & Engka, J. N. A. (2021). Latihan Fisik Pada Pasien Obesitas. *JURNAL BIOMEDIK (JBM)*, 13(1), 34.
<https://doi.org/10.35790/jbm.13.1.2021.31773>

Setiyani, L., Nur Indahsari, A., & Roestam, R. (2023). Analisis Prediksi Level Obesitas Menggunakan Perbandingan Algoritma Machine Learning dan Deep Learning. *Jurnal Teknologi Rekayasa*, 8(1), 139–146.
<https://doi.org/10.31544/jtera.v8.i1.2023.139-146>

Tasari, A., Dinata Tarigan, D., Nia, E., Purba, D. B., Saputra, K., & Artikel, I. (2022). Perbandingan Algoritma Support Vector Machine dan KNN dalam Memprediksi Struktur Sekunder Protein. *JURNAL INFORMATIKA*, 9(2).
<https://raw.githubusercontent.com/JiayingLi/Pr>

LAMPIRAN

1. Keterangan Turnitin



UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN
Gedung Rektorat Lantai 3,
Jl. Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

SURAT KETERANGAN HASIL PENGECEKAN TURNITIN

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : PERBANDINGAN ALGORITMA KNN DAN SVM DALAM
PENGEMBANGAN MODEL PREDIKSI OBESITAS PADA
DESTROYER MUSCLE GYM

Penulis : Anandina Khairunnisa

NIM : 2020230050

Tgl pemeriksaan : 17 Juli 2024

Dengan hasil Tingkat Kesamaan (*similarity index*) 21%

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 17 Juli 2024

Ka.UPT Perpustakaan Unsada

Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi
Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan dan Pasca Sarjana

2020230050_Anandina Khairunnisa_PERBANDINGAN
ALGORITMA KNN DAN SVM DALAM PENGEMBANGAN MODEL
PREDIKSI OBESITAS PADA DESTROYER MUSCLE GYM

ORIGINALITY REPORT



PRIMARY SOURCES

1	jurnal.polsri.ac.id Internet Source	1%
2	www.slideshare.net Internet Source	1%
3	docplayer.info Internet Source	1%
4	kc.umn.ac.id Internet Source	1%
5	garuda.kemdikbud.go.id Internet Source	1%
6	jikm.upnvj.ac.id Internet Source	1%
7	repository.ub.ac.id Internet Source	1%
8	repositori.unsil.ac.id Internet Source	1%

pdffox.com

2. Model.ipnyb

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_recall_fscore_support
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

df_klasifikasi_train = pd.read_csv("ObesityDataSet_Kaggle.csv")
df_klasifikasi_test = pd.read_csv("Testing.csv")

X_train = df_klasifikasi_train[['Gender', 'Age', 'Height',
'Weight']]
y_train = df_klasifikasi_train['Label']
X_test_new = df_klasifikasi_test[['Gender', 'Age', 'Height',
'Weight']]
y_test_new = df_klasifikasi_test['Label']

numerical_features = ['Age', 'Height', 'Weight']
categorical_features = ['Gender']
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(), categorical_features)])

# KNN
knn_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                ('classifier',
                                KNeighborsClassifier(n_neighbors=5))])
knn_model = knn_pipeline.fit(X_train, y_train)
y_pred_new = knn_pipeline.predict(X_test_new)
cm_new = confusion_matrix(y_test_new, y_pred_new)

plt.figure(figsize=(16, 6))
plt.imshow(cm_new, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()
classes = y_test_new.unique()
tick_marks = range(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)
for i in range(cm_new.shape[0]):
    for j in range(cm_new.shape[1]):
        plt.text(j, i, format(cm_new[i, j], 'd'),
                 horizontalalignment="center",
```

```

        color="white" if cm_new[i, j] > cm_new.max() / 2
else "black")
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.tight_layout()
plt.show()

accuracy_new = accuracy_score(y_test_new, y_pred_new)
print('Akurasi Testing KNN :', accuracy_new)
P_new = precision_recall_fscore_support(y_test_new, y_pred_new,
average='macro')
print("Precision : ", P_new[0])
print("Recall : ", P_new[1])
print("F Score : ", P_new[2])

# SVM
svm_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                              ('classifier', SVC(kernel='linear',
random_state=42))])
svm_model = svm_pipeline.fit(X_train, y_train)
y_pred_new = svm_pipeline.predict(X_test_new)
cm_new = confusion_matrix(y_test_new, y_pred_new)

plt.figure(figsize=(16, 6))
plt.imshow(cm_new, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()
classes = y_test_new.unique()
tick_marks = range(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)

for i in range(cm_new.shape[0]):
    for j in range(cm_new.shape[1]):
        plt.text(j, i, format(cm_new[i, j], 'd'),
                horizontalalignment="center",
                color="white" if cm_new[i, j] > cm_new.max() / 2
else "black")
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.tight_layout()
plt.show()

accuracy_new = accuracy_score(y_test_new, y_pred_new)
print('Akurasi Testing SVM :', accuracy_new)

P_new = precision_recall_fscore_support(y_test_new, y_pred_new,
average='macro')
print("Precision : ", P_new[0])

```



```
print("Recall : ", P_new[1])
print("F Score : ", P_new[2])

import pickle
# Latih model SVM pada dataset latih{
svm_model = svm_pipeline.fit(X_train, y_train)
# Simpan model ke file menggunakan pickle
with open('svm_model.pkl', 'wb') as model_file:
    pickle.dump(svm_model, model_file)
```