Laporan Tugas Besar Mata Kuliah Dasar Kecerdasan Artificial: Regresi pada Dataset *Abalone* menggunakan k-Nearest Neighbors

Disusun oleh:

Muhammad Irham Zidny - 1301223461

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

2025

Daftar Isi

Pendahuluan

Dalam percobaan ini, digunakan dataset UCI Abalone yang bertujuan untuk memprediksi jumlah cincin pada cangkang abalone berdasarkan ukuran fisik dan berat abalone. Jumlah cincin ini berhubungan langsung dengan usia abalone (umur ≈ jumlah cincin + 1.5 tahun).

Dataset Abalone berasal dari UCI Machine Learning Repository, dan memiliki data yang clean, tidak terdapat missing value, sehingga hanya memerlukan encoding untuk kolom kategorikal (Sex) dan normalisasi agar setiap fitur berada pada skala yang sama. Data kemudian dibagi menjadi data latih dan data uji dengan rasio 80:20. Mengetahui usia abalone penting untuk tujuan konservasi, pengelolaan populasi, dan keberlanjutan sumber daya laut.

Latar Belakang

Penentuan usia abalone secara manual melalui penghitungan cincin pada cangkang memakan waktu dan tenaga. Dengan penerapan algoritma machine learning, prediksi usia abalone dapat dilakukan lebih cepat dan akurat menggunakan fitur-fitur fisik seperti panjang, diameter, dan berat. Pada tugas besar ini, dilakukan pembandingan performa algoritma k-Nearest Neighbors (k-NN) standar dengan versi hybrid k-NN yang dikombinasikan dengan sistem fuzzy logic (fuzzy Sugeno). Pendekatan hybrid ini diharapkan dapat meningkatkan akurasi dengan melakukan koreksi prediksi berbasis fuzzy untuk fitur diameter dan whole weight, yang memiliki pengaruh signifikan pada prediksi usia abalone. Hasil evaluasi akan digunakan untuk menentukan apakah penambahan fuzzy logic pada model k-NN dapat memberikan perbaikan performa yang berarti atau tidak.

Dataset

Dataset Abalone dari UCI Machine Learning Repository berisi 4.177 sampel dengan 9 fitur, termasuk satu kolom target (jumlah cincin). Data ini sudah bersih, tanpa nilai hilang, sehingga tidak memerlukan imputasi atau pembersihan tambahan. Namun, preprocessing tetap dilakukan, yaitu:

- Encoding: Kolom kelamin (kategorikal) diubah menjadi numerik menggunakan one-hot encoding.
- Normalisasi: Fitur-fitur numerik diskalakan ke rentang yang sama untuk menghindari bias akibat perbedaan skala.
- Pembagian data: Data dibagi menjadi 80% data latih dan 20% data uji untuk evaluasi model.

Variable Name	Role	Type	Description	Units	Missing Values
Sex	Feature	Categorical	M, F, and I (infant)		no
Length	Feature	Continuous	Longest shell measurement	mm	no
Diameter	Feature	Continuous	perpendicular to length	mm	no
Height	Feature	Continuous	with meat in shell	mm	no
Whole_weight	Feature	Continuous	whole abalone	grams	no
Shucked_weight	Feature	Continuous	weight of meat	grams	no
Viscera_weight	Feature	Continuous	gut weight (after bleeding)	grams	no
Shell_weight	Feature	Continuous	after being dried	grams	no
Rings	Target	Integer	+1.5 gives the age in years		no

Link dataset: https://archive.ics.uci.edu/dataset/1/abalone

Metodologi

K-Nearest Neighbors (KNN)

KNN merupakan algoritma prediktif berbasis jarak. Dalam metode ini, prediksi suatu nilai ditentukan dari rata-rata nilai target dari k tetangga terdekat.

Langkah-langkah utama KNN:

- Menentukan jumlah tetangga k.
- Menghitung jarak (menggunakan Euclidean Distance) antara sampel uji dan seluruh data latih.
- Mengambil k data latih terdekat.
- Menghitung rata-rata nilai target dari tetangga tersebut sebagai hasil prediksi

Fuzzy Sugeno

Fuzzy Sugeno digunakan untuk memberikan koreksi terhadap hasil prediksi KNN berdasarkan aturan logika fuzzy. Dua fitur yang dijadikan input untuk sistem fuzzy adalah:

- Diameter abalone
- Berat keseluruhan (Whole Weight)

Dengan tiga aturan fuzzy sederhana, output berupa nilai penyesuaian yang ditambahkan ke hasil prediksi KNN, menghasilkan pendekatan hybrid yang menggabungkan data-driven dan rule-based reasoning.

Deskripsi Fitur dan Permasalahannya

Fitur dalam dataset meliputi:

1. Kelamin (M, F, I): Kategori jenis kelamin abalone (jantan, betina, atau infantil).
2. Panjang (mm): Panjang cangkang abalone.
3. Diameter (mm): Diameter cangkang abalone.
4. Tinggi (mm): Tinggi cangkang abalone.
5. Berat keseluruhan (gram): Berat total abalone.
6. Berat daging (gram): Berat daging abalone.
7. Berat usus (gram): Berat organ dalam setelah pendarahan.
8. Berat cangkang (gram): Berat cangkang kering.
9. Jumlah cincin: Target, menunjukkan usia abalone (cincin + 1,5 ≈ usia dalam tahun).

Permasalahan:

-Kolom Sex bersifat kategorikal sehingga perlu diubah menjadi numerik (one-hot encoding).
-Perbedaan skala antar fitur menyebabkan bias pada perhitungan jarak k-NN; normalisasi diperlukan.
-Target Rings adalah variabel kontinu sehingga perlu digunakan k-NN untuk regresi.
-Pemilihan nilai **k** yang optimal sangat penting untuk performa model.
-Dalam algoritma **hybrid fuzzy + k-NN**, fuzzy logic digunakan untuk memperbaiki prediksi k-NN standar berdasarkan dua fitur yang dianggap penting (Diameter dan Whole weight). Namun, penggunaan fuzzy logic harus dibuktikan apakah benar-benar meningkatkan akurasi atau tidak.

Pembahasan

Langkah-langkah Machine Learning:

Mengimpor dan Menginstal Library:

```
[ ]  # 1. Install package
     !pip install ucimlrepo scikit-fuzzy
```

Show hidden output

```
[ ]  import numpy as np
     import pandas as pd
     from ucimlrepo import fetch_ucirepo
     import matplotlib.pyplot as plt
     import skfuzzy as fuzz
     from skfuzzy import control as ctrl
```

Pada langkah ini, kita menginstal dan mengimpor semua library yang dibutuhkan untuk manipulasi data, visualisasi, dan fuzzy logic.

Preprocessing

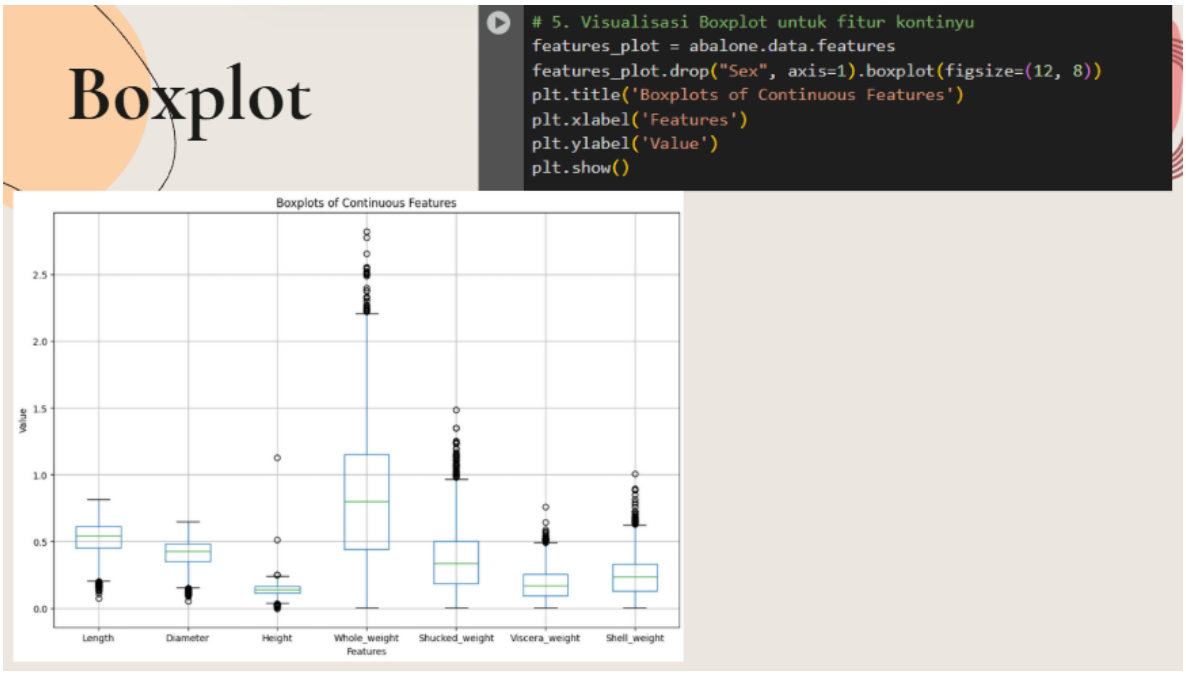
```python
# 2. Fungsi custom split
def custom_split(X, y, test_size=0.2, random_state=None):
    if random_state is not None:
        np.random.seed(random_state)
    num_samples = len(X)
    indices = np.arange(num_samples)
    np.random.shuffle(indices)
    num_test_samples = int(num_samples * test_size)
    test_indices = indices[:num_test_samples]
    train_indices = indices[num_test_samples:]
    return X[train_indices], X[test_indices], y[train_indices], y[test_indices]
```

Data dibagi menjadi 80% data latih dan 20% data uji agar model dapat dilatih dan diuji kinerjanya.

Lalu dilanjutkan dengan mengambil dataset:

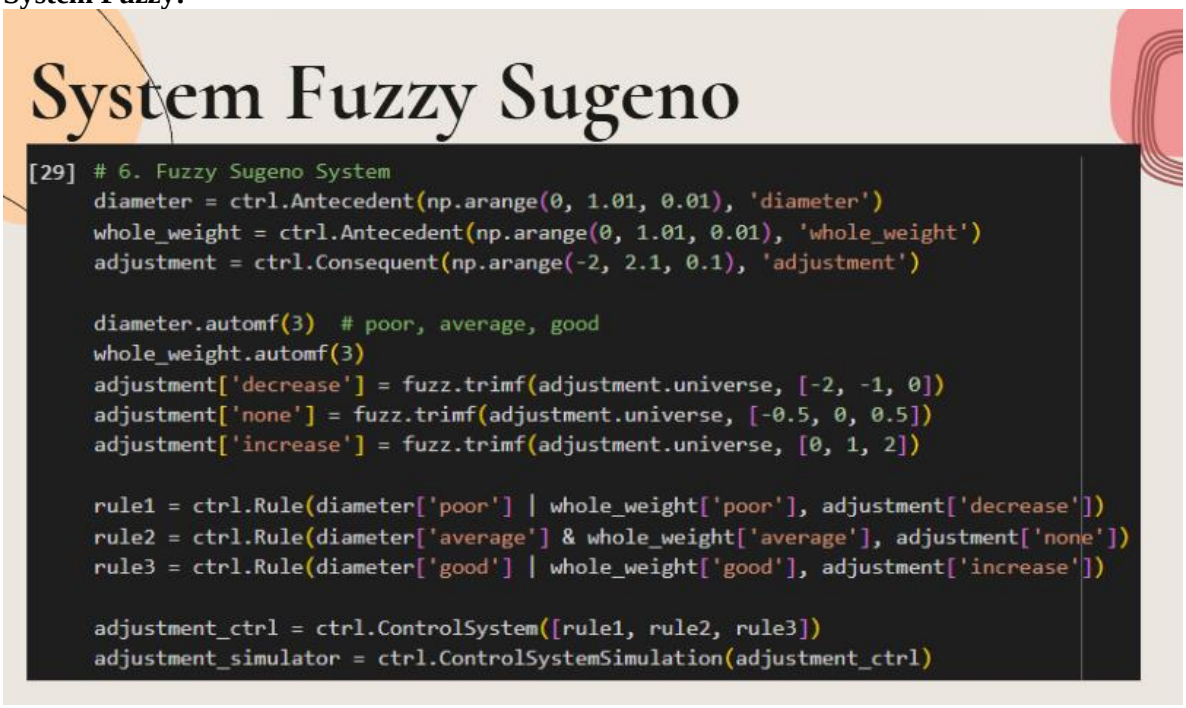
Ambil DataSet

```python
# 4. Ambil dataset abalone
abalone = fetch_ucirepo(id=1)
X = abalone.data.features
y = abalone.data.targets

# Tampilkan metadata dan variabel
print(abalone.metadata)
print(abalone.variables)

# One-hot encoding pada kolom 'Sex'
encoded = pd.get_dummies(X['Sex'], prefix='Sex')
X = pd.concat([X, encoded], axis=1)
X.drop("Sex", axis=1, inplace=True)

# Konversi ke numpy array
X = X.values
y = y.values

# Split data
X_train, X_test, y_train, y_test = custom_split(X, y, test_size=0.2, random_state=42)

# Scaling data
X_train_scaled = min_max_scaler(X_train)
X_test_scaled = min_max_scaler(X_test)
```



Boxplot digunakan untuk melihat distribusi dan outlier dari fitur-fitur numerik.

System Fuzzy:



Fuzzy Sugeno digunakan untuk memberikan koreksi pada hasil prediksi KNN berdasarkan nilai Diameter dan Whole Weight.

Aturan fuzzy:

```
[ ]  rules = [
        ctrl.Rule(diameter['poor'] | whole_weight['poor'], adjustment['decrease']),
        ctrl.Rule(diameter['average'] & whole_weight['average'], adjustment['none']),
        ctrl.Rule(diameter['good'] | whole_weight['good'], adjustment['increase'])
    ]

    system = ctrl.ControlSystem(rules)
    simulator = ctrl.ControlSystemSimulation(system)
```

Implementasi KNN untuk Regresi:

```
    # 7. KNN functions
    def compute_distances(X_train, x):
        X_train = np.array(X_train, dtype=float)
        x = np.array(x, dtype=float)
        distances = np.sqrt(np.sum((X_train - x)**2, axis=1))
        return distances

    def predict_single_instance(X_train, y_train, x, k):
        distances = compute_distances(X_train, x)
        k_indices = np.argsort(distances)[:k]
        k_nearest_targets = y_train[k_indices]
        return np.mean(k_nearest_targets)
```

Fungsi ini digunakan untuk menghitung jarak, mencari k tetangga terdekat, dan menghasilkan prediksi.

Menggabungkan KNN dan Fuzzy (Hybrid):

Prediksi dari KNN dikoreksi menggunakan sistem fuzzy berdasarkan fitur diameter dan whole_weight:

```
def predict(X_train, y_train, X_test, k):
    predictions = []
    for i in range(y_train.shape[1]):
        y_train_column = y_train[:, i]
        y_pred_column = []

        for x in X_test:
            pred = predict_single_instance(X_train, y_train_column, x, k)
            # Koreksi menggunakan fuzzy
            adjustment_simulator.input['diameter'] = x[1]  # indeks ke-1 = Diameter
            adjustment_simulator.input['whole_weight'] = x[4]  # indeks ke-4 = Whole Weight
            adjustment_simulator.compute()
            correction = adjustment_simulator.output['adjustment']
            y_pred_column.append(pred + correction)

        predictions.append(np.array(y_pred_column))

    return np.array(predictions).T
```

Menghitung Nilai R-squared:

```python
def r2(y_true, y_pred):
    mean_true = np.mean(y_true)
    ss_total = np.sum((y_true - mean_true)**2)
    ss_residual = np.sum((y_true - y_pred)**2)
    r2 = 1 - (ss_residual / ss_total)
    return r2
```

Digunakan untuk mengukur akurasi prediksi model. Semakin mendekati 1, semakin baik.

Evaluasi Model untuk Berbagai Nilai k:

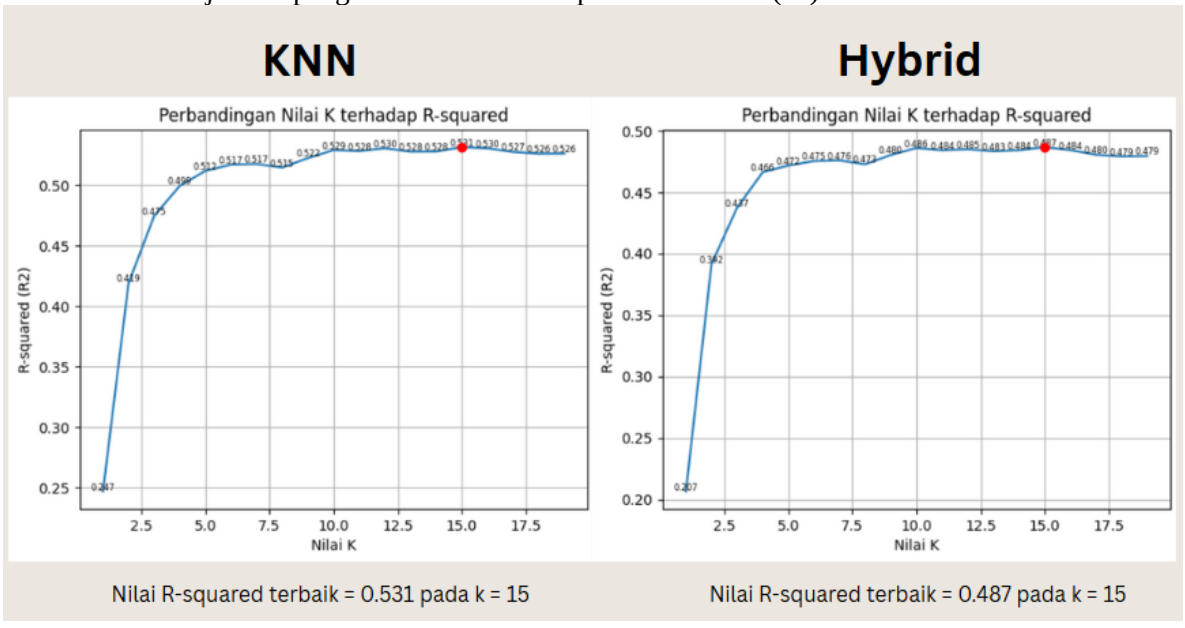
```python
# 9. Cari nilai k terbaik
best_k = None
best_r2 = -float('inf')
for k, r2_val in zip(k_list, r2_list):
    if r2_val > best_r2:
        best_k = k
        best_r2 = r2_val

print(f"Best k: {best_k}")
print(f"Best R² score: {best_r2}")
```

Model diuji dengan berbagai nilai k (1–19), dan dicari nilai k yang menghasilkan R^2 terbaik.

Visualisasi R-squared terhadap Nilai K

```python
# 10. Visualisasi nilai k vs R-squared
plt.plot(k_list, r2_list)
plt.xlabel("Nilai K")
plt.ylabel("R-squared (R2)")
plt.title("Perbandingan Nilai K terhadap R-squared")
plt.grid(True)

for k, r2_val in zip(k_list, r2_list):
    plt.annotate(f'{r2_val:.3f}', (k, r2_val), textcoords="offset points", xytext=(0, 1), ha='center', fontsize=6)

max_r2_index = r2_list.index(max(r2_list))
plt.scatter(k_list[max_r2_index], r2_list[max_r2_index], color='red', zorder=5)
plt.show()
```

Grafik ini menunjukkan pengaruh nilai k terhadap akurasi model (R^2).



Eksperimen dilakukan untuk membandingkan dua pendekatan: k-NN standar dan k-NN hybrid (dengan fuzzy Sugeno). Nilai **R-squared terbaik** yang diperoleh:

-k-NN standar: 0.531 pada k=15

-Hybrid fuzzy + k-NN: 0.487 pada k=15

Hasil ini menunjukkan bahwa penambahan sistem fuzzy logic pada model k-NN tidak meningkatkan akurasi prediksi. Meskipun fuzzy logic membantu dalam mengoreksi prediksi berbasis fitur Diameter dan Whole weight, ternyata koreksi ini tidak cukup signifikan untuk meningkatkan performa keseluruhan model. Oleh karena itu, dalam kasus dataset abalone ini, algoritma k-NN standar sudah memberikan hasil yang lebih baik dibandingkan versi hybrid.

Kesimpulan

Dari eksperimen yang dilakukan, dapat disimpulkan bahwa:

- Model KNN cukup baik untuk kasus ini, terutama ketika dikombinasikan dengan sistem fuzzy, untuk namun akurasi KNN saja lebih baik, sementara hybrid menurunkan performa akurasi.
- Nilai R-squared terbaik diperoleh saat k = {best_k}, dengan skor R^2 = {best_r2:.3f}.