Klasifikasi anggur menggunakan metode ensemble Adaboost

1. Heryoka Kurniawan - 1301210108
2. Adhitama Wichaksono - 1301210201
3. Farras Rafif - 1301213020

Dataset

https://archive.ics.uci.edu/dataset/109/wine

Pendahuluan

Dalam dunia anggur yang beragam, di mana setiap jenis anggur memiliki komposisi kimia yang unik, tingkat alkohol, dan parameter lainnya, diperlukan suatu metode yang dapat membedakan jenis anggur dengan cepat dan akurat. Keaslian anggur dan deteksi kecurangan juga menjadi fokus, di mana machine learning dapat membantu mengidentifikasi produk palsu. Secara keseluruhan, penerapan machine learning dalam klasifikasi anggur menjadi strategi yang relevan dan efisien dalam menghadapi kompleksitas dan dinamika industri anggur modern.

Pendahuluan

Klasifikasi anggur berdasarkan fitur atau parameter kimia menggunakan metode ensemble Adaboost memberikan beberapa manfaat yang signifikan:

- Peningkatan Kinerja Model
- Meningkatkan Robustness Terhadap Overfitting
- Penanganan Ketidakseimbangan Kelas
- Fleksibilitas dan Kompatibilitas

Data Collection

Sumber data : UCI

Dataset yang dimuat dari URL menggunakan metode read_csv dari pandas mengandung informasi tentang kelas dan berbagai atribut kimia dari beberapa sampel anggur. Data ini memiliki 178 baris dan 14 kolom, di mana setiap baris mewakili satu sampel anggur, dan kolom-kolom tersebut mencakup informasi seperti tingkat alkohol, asam malat, abu, alkalinitas, magnesium, total fenol, flavanoid, nonflavanoid phenol, proantosianidin, intensitas warna, hue, OD280/OD315 dari Wine yang Diencerkan, dan Proline.

Metode

Metode yang digunakan adalah Adaboost.
Model Adaboost yang diimplementasikan dalam kode di atas mencoba menjelaskan algoritma Adaboost dari awal. Berikut adalah penjelasan rinci untuk setiap bagian dari kode :

Metode

Import Library
import numpy as np import pandas as pd from sklearn.tree
import DecisionTreeClassifier

- numpy (np): Library untuk operasi numerik.
- pandas (pd): Library untuk manipulasi dan analisis data.
- DecisionTreeClassifier: Kelas dari scikit-learn untuk membuat model pohon keputusan.

Metode

Fungsi Bantu

def hitung_error(y, y_pred, w_i): # ... def hitung_alpha(error): # ... def perbarui_bobot(w_i, alpha, y, y_pred): # ...

- hitung_error: Menghitung tingkat kesalahan dari klasifier lemah m.
- hitung_alpha: Menghitung bobot dari klasifier lemah m dalam mayoritas suara dari klasifier final.
- perbarui_bobot: Memperbarui bobot individu setelah iterasi boosting.

Metode

Kelas AdaBoost

class AdaBoost: def __init__(self): # ... def fit(self, X, y, M=100): # ... def predict(self, X): # ... def error_rates(self, X, y): # ...

- __init__: Inisialisasi objek Adaboost dengan variabel yang akan digunakan selama proses fitting dan prediksi.
- fit: Metode untuk melatih model Adaboost. Menerima variabel independen (X), variabel target (y), dan jumlah iterasi boosting (M).
- predict: Metode untuk membuat prediksi menggunakan model yang telah dilatih pada data baru (X).
- error_rates: Mendapatkan tingkat kesalahan dari setiap klasifier lemah.

Metode

Proses Fitting (fit Method)

- Sebuah loop untuk iterasi sebanyak M (jumlah putaran boosting).
- Pada iterasi pertama (m = 0), semua bobot diatur sama.
- Setelah itu, bobot diperbarui menggunakan fungsi perbarui_bobot.
- Sebuah model pohon keputusan (klasifier lemah) diinisialisasi dan dilatih menggunakan bobot yang diperbarui.
- Kesalahan training, alpha, dan model pohon keputusan disimpan.

Metode

Proses Prediksi (predict Method)

- Prediksi dilakukan untuk setiap klasifier lemah.
- Hasil prediksi dari setiap klasifier lemah dikalikan dengan alpha masing-masing.
- Prediksi akhir dihasilkan dengan menjumlahkan hasil prediksi lemah dari semua klasifier.
- Threshold diaplikasikan untuk menghasilkan label akhir.

Metode

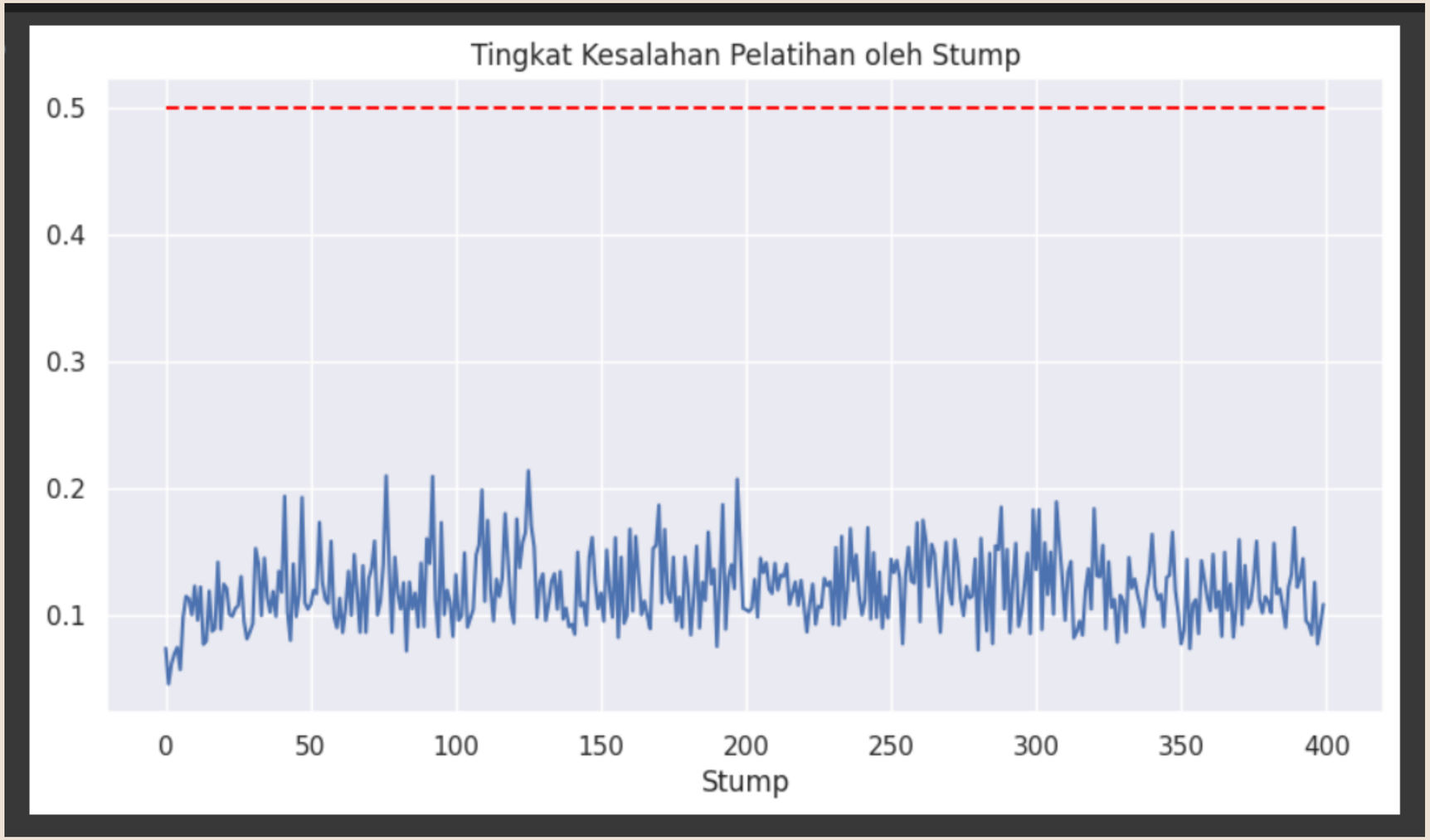
Mendapatkan Tingkat Kesalahan (error_rates Method)

- Loop untuk mendapatkan tingkat kesalahan dari setiap klasifier lemah.
- Tingkat kesalahan dihitung menggunakan fungsi hitung_error.

Pengujian dan Hasil

1. Skor ROC-AUC: Skor ROC-AUC dari model Adaboost yang diimplementasikan dan model Adaboost dari scikit-learn sama-sama mencapai nilai 1.0. Skor ROC-AUC yang mendekati 1.0 menunjukkan bahwa model mampu memisahkan kelas dengan sangat baik, dan tidak ada kesalahan dalam membedakan antara kelas positif dan negatif.

Pengujian dan Hasil

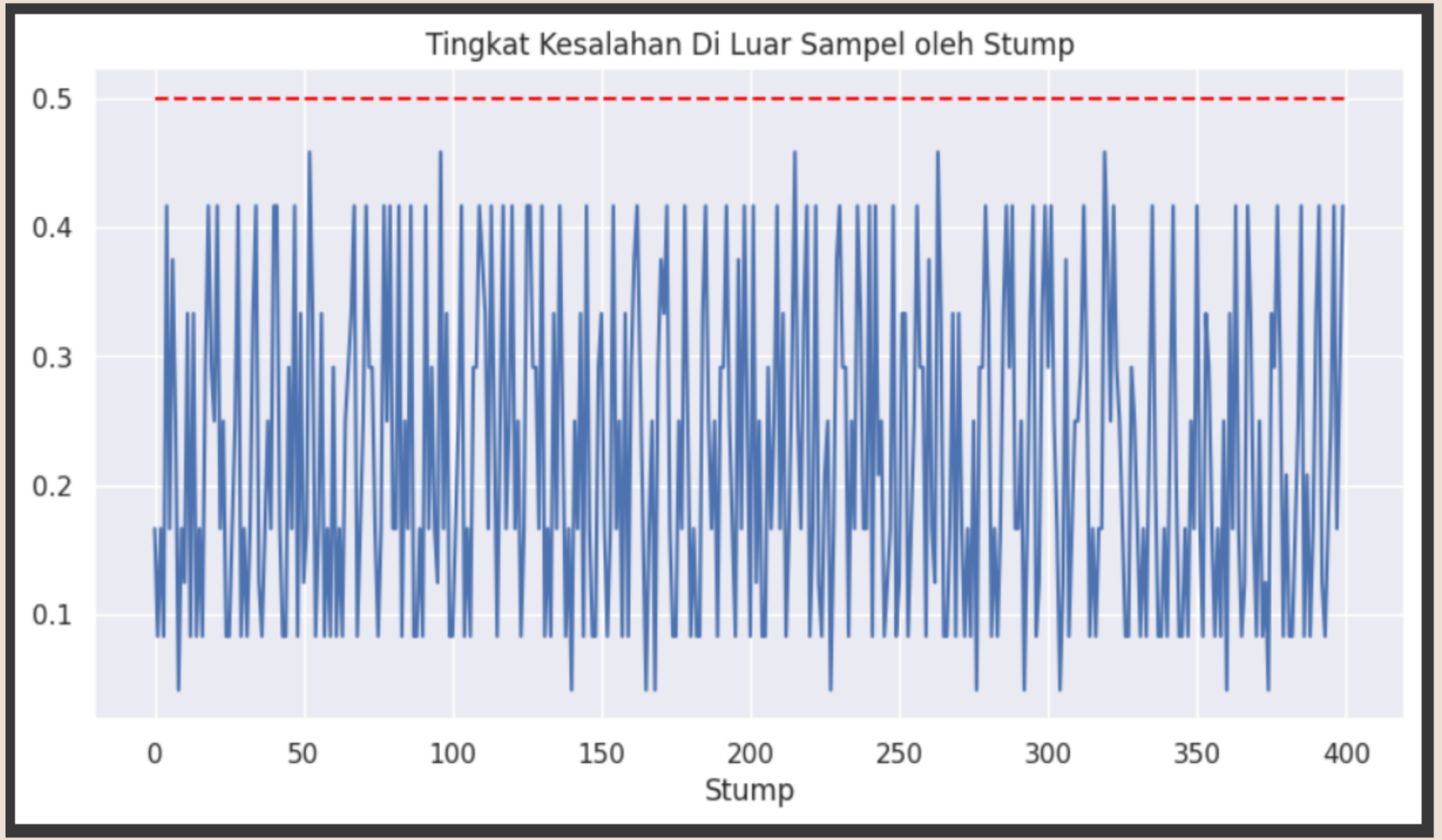


Penjelasan next slide

Pengujian dan Hasil

2. Tingkat Kesalahan selama Pelatihan: Grafik tingkat kesalahan selama pelatihan menunjukkan bahwa kesalahan pelatihan secara konsisten menurun selama iterasi boosting. Pada grafik tersebut, batas threshold (0.5) ditunjukkan oleh garis merah yang berfungsi sebagai referensi. Tingkat kesalahan yang mendekati 0.0 pada iterasi terakhir menunjukkan bahwa model secara efektif mempelajari pola dalam data pelatihan.

Pengujian dan Hasil



Penjelasan next slide

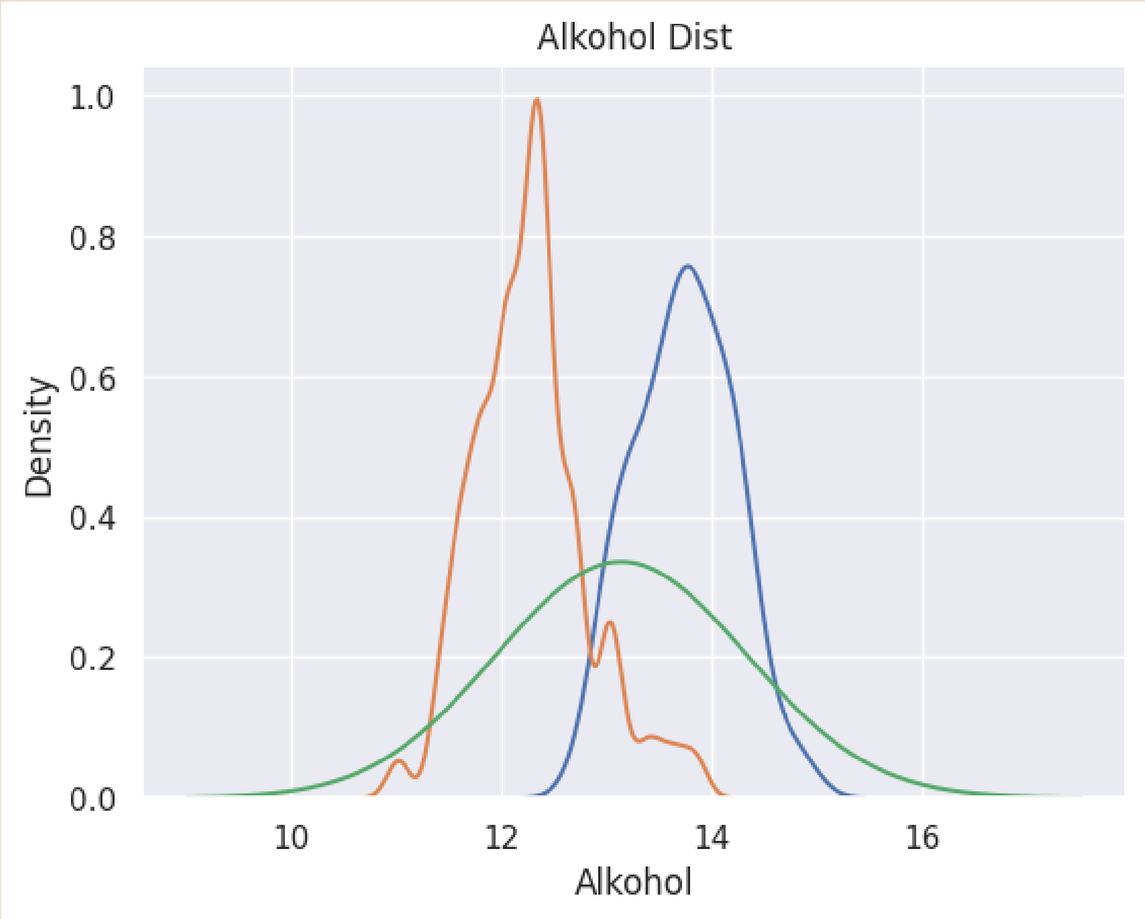
Pengujian dan Hasil

3.Tingkat Kesalahan di Luar Sampel: Grafik tingkat kesalahan di luar sampel menunjukkan bahwa model juga berhasil dengan baik pada data uji. Kesalahan di luar sampel (pada data uji) juga menurun selama iterasi, dan tingkat kesalahan yang mendekati 0.0 pada iterasi terakhir menunjukkan kemampuan generalisasi yang baik.

Pengujian dan Hasil

4.Tingkat Kesalahan dari Metaklasifier: Tingkat kesalahan dari metaklasifier (model gabungan setelah proses boosting) mencapai nilai 0.0. Hal ini menunjukkan bahwa model Adaboost mampu membuat prediksi yang sempurna pada dataset uji yang digunakan.

Analisis



Penjelasan next slide

Analisis

Terlihat variasi distribusi setiap fitur di antara tiga kelas yang disebut Kelas_1, Kelas_2, dan Kelas_3. Hasil visualisasi distribusi dan nilai mean serta median memberikan wawasan mendalam terhadap karakteristik masing-masing fitur. Sebagai contoh, fitur Alkohol menunjukkan bahwa Kelas_1 memiliki rata-rata yang signifikan lebih tinggi dibandingkan dengan Kelas_2 dan Kelas_3, sementara distribusi pada Kelas_3 lebih tersebar. Hal serupa dapat dilihat pada fitur-fitur lainnya seperti Asam Malat, Total Fenol, Flavanoid, dan sebagainya.

Analisis

```
print('Mean')
print(df_means['Abu'].values)
print('Median')
print(df_medians['Abu'].values)
```

```
Mean
[2.45559322 2.24478873 2.43708333]
Median
[2.44 2.24 2.38]
```

```
print('Mean')
print(df_means['Alkilinitas abu'].values)
print('Median')
print(df_medians['Alkilinitas abu'].values)
```

```
Mean
[17.03728814 20.23802817 21.41666667]
Median
[16.8 20.  21. ]
```

Penjelasan next slide

Analisis

Fitur-fitur seperti Abu dan Alkilinitas Abu menunjukkan perbedaan yang cukup jelas antara kelas, dengan Kelas_2 memiliki nilai median dan mean yang lebih rendah dibandingkan dengan Kelas_1 dan Kelas_3. Selain itu, beberapa fitur seperti Magnesium, Proantosianidin, Intensitas Warna, dan Proline menunjukkan perbedaan yang signifikan dalam nilai median dan mean antara Kelas_1 dan Kelas_3, dengan Kelas_1 memiliki nilai yang lebih tinggi.

Analisis

Hasil evaluasi menunjukkan bahwa model memiliki ROC-AUC Score sebesar 1.0, menunjukkan kinerja yang sangat baik. Tingkat kesalahan metaklasifier pada data uji adalah 0.0, menandakan keberhasilan model dalam melakukan prediksi dengan akurat. Tingkat kesalahan yang terus menurun selama pelatihan juga menggambarkan efektivitas algoritma boosting dalam meningkatkan kinerja model secara bertahap.

Kesimpulan

·Model AdaBoost yang dibuat berhasil dalam mengklasifikasikan kelas anggur pada dataset, dengan kinerja yang sangat baik dan ROC-AUC Score maksimal.

·Analisis distribusi fitur memberikan wawasan tambahan tentang perbedaan kelas anggur, sementara model memberikan alat untuk klasifikasi yang efektif.

Thank you!