Laporan Project Based CLO 4 Machine Learning

“Implementation of Ensemble Method on Classification Task”

Disusun oleh :

1. Heryoka Kurniawan - 1301210108

2. Adhitama Wichaksono - 1301210201

3. Farras Rafif - 1301213020

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. Kompetisi yang ketat di industri anggur mendorong produsen untuk mencari cara inovatif untuk membedakan produk mereka, dan pemahaman yang lebih mendalam tentang komposisi anggur dapat memberikan keunggulan kompetitif. Selain itu, pentingnya penilaian kualitas anggur, pemantauan proses produksi dengan teknologi sensor, serta pertumbuhan minat pada analisis data dan kecerdasan buatan (AI) semakin mendorong penerapan machine learning dalam klasifikasi anggur. Keaslian anggur dan deteksi kecurangan juga menjadi fokus, di mana machine learning dapat membantu mengidentifikasi produk palsu. Dengan kemajuan teknologi pengolahan citra, visualisasi etiket anggur juga dapat dimanfaatkan untuk klasifikasi. Secara keseluruhan, penerapan machine learning dalam klasifikasi anggur menjadi strategi yang relevan dan efisien dalam menghadapi kompleksitas dan dinamika industri anggur modern.

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

1. **Peningkatan Kinerja Model:**

   Adaboost adalah metode ensemble yang menggabungkan hasil dari beberapa model lemah (weak learners) untuk membentuk model yang kuat. Dalam konteks klasifikasi anggur, fitur-fitur kimia dapat dianggap sebagai indikator yang dapat digunakan untuk mengenali karakteristik unik dari setiap jenis anggur. Dengan menggabungkan hasil dari beberapa model, Adaboost dapat meningkatkan kemampuan model untuk mengklasifikasikan anggur dengan akurasi yang lebih tinggi.

2. **Meningkatkan Robustness Terhadap Overfitting:**

   Adaboost cenderung lebih tahan terhadap overfitting daripada model tunggal. Overfitting terjadi ketika model terlalu kompleks dan "menghafal" data pelatihan sehingga tidak dapat menggeneralisasi dengan baik pada data baru. Adaboost, dengan cara menggabungkan hasil dari model lemah, dapat membantu mengurangi risiko overfitting dan meningkatkan kemampuan model untuk melakukan generalisasi pada data yang tidak terlihat sebelumnya.

3. **Penanganan Ketidakseimbangan Kelas:**

   Jika data anggur yang digunakan untuk pelatihan memiliki ketidakseimbangan antara kelas (misalnya, satu jenis anggur mungkin lebih banyak daripada yang lain), Adaboost dapat membantu mengatasi masalah ini. Metode ini memberikan lebih banyak bobot pada contoh-contoh yang salah diklasifikasikan oleh model sebelumnya, sehingga fokus pada memperbaiki kesalahan tersebut.

4. **Fleksibilitas dan Kompatibilitas:**

   Adaboost dapat digunakan dengan berbagai jenis model lemah, seperti pohon keputusan (decision trees), yang dapat cocok untuk data yang kompleks dan tidak linier. Ini memberikan fleksibilitas dalam memilih model yang paling sesuai dengan karakteristik data anggur.

5. **Interpretabilitas:**

   Pohon keputusan, yang sering digunakan sebagai weak learner dalam Adaboost, memiliki interpretabilitas yang baik. Ini berarti hasil dari model dapat dijelaskan dengan relatif mudah, membantu pemahaman tentang faktor apa yang mempengaruhi klasifikasi anggur.

6. **Penyebaran Kesalahan (Error Disaggregation):**

   Adaboost dapat memberikan informasi tentang seberapa baik model berkinerja pada setiap contoh data. Ini memungkinkan untuk memahami lebih baik di mana dan mengapa model mungkin membuat kesalahan, yang dapat digunakan untuk perbaikan lebih lanjut atau untuk mendapatkan wawasan tambahan tentang data.

Dengan menggabungkan keuntungan-keuntungan ini, klasifikasi anggur menggunakan metode ensemble Adaboost dapat memberikan hasil yang lebih kuat dan dapat diandalkan.

Dataset

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.

Awalnya, judul kolom tidak benar, dan koreksi diberikan dengan menggunakan variabel column_headers. Setelah itu, dataset diproses dan dijelajahi dengan menggunakan beberapa metode pandas seperti isnull(), describe(), dan groupby() untuk memahami statistik deskriptif dan distribusi data.

Selanjutnya, dataset dibagi berdasarkan kelas anggur (Kelas 1, Kelas 2, Kelas 3), dan rata-rata serta median dari setiap kelas dihitung untuk setiap atribut. Visualisasi distribusi atribut-atribut ini dilakukan menggunakan kernel density estimation (KDE) plots.

Hasil analisis statistik dan visualisasi memberikan wawasan tentang bagaimana atribut-atribut ini berbeda antar kelas anggur, membantu pemahaman lebih lanjut tentang karakteristik setiap kelas. Sebagai contoh, distribusi atribut seperti tingkat alkohol, asam malat, dan magnesium dapat dibandingkan antar kelas menggunakan KDE plots dan statistik rata-rata serta median.

Fitur-fitur tersebut memiliki makna sebagai berikut:

1. **Kelas**: Ini adalah kolom yang menunjukkan kategori atau kelas dari suatu sampel atau baris data. Pada dataset ini, kelompok kelas dapat bernilai 1, 2, atau 3, dan mungkin mencerminkan kelompok atau kategori tertentu dari contoh anggur.
2. **Alkohol**: Ini adalah kolom yang mencatat kadar alkohol dalam sampel anggur, diukur dalam persentase.
3. **Asam Malat**: Menunjukkan tingkat asam malat dalam anggur. Asam malat adalah salah satu jenis asam organik dalam anggur.
4. **Abu**: Ini adalah kolom yang mencatat jumlah abu yang terkandung dalam sampel anggur.
5. **Alkilinitas Abu**: Menunjukkan tingkat alkilinitas abu dalam anggur. Alkilinitas abu mencerminkan sejauh mana abu dapat mempertahankan tingkat keasaman dalam anggur.
6. **Magnesium**: Mencatat jumlah magnesium dalam sampel anggur.
7. **Total Fenol**: Ini adalah kolom yang mencatat jumlah total fenol dalam anggur. Fenol adalah senyawa kimia yang dapat memberikan rasa dan warna tertentu pada anggur.
8. **Flavanoid**: Merupakan kelas senyawa fenolik tertentu yang terdapat dalam anggur.
9. **Nonflavanoid Phenol**: Ini adalah kolom yang mencatat jumlah senyawa fenolik nonflavanoid dalam anggur.

10. **Proantosianidin**: Menunjukkan jumlah proantosianidin dalam anggur, yang merupakan kelompok senyawa fenolik.
11. **Intensitas Warna**: Mengukur intensitas warna dalam anggur.
12. **Hue**: Menunjukkan tingkat warna pada skala hue.
13. **OD280/OD315 dari Wine yang Diencerkan**: Rasio optik dari dua panjang gelombang, yang sering digunakan untuk mengukur warna dalam anggur.
14. **Proline**: Merupakan suatu parameter yang diukur dalam proses analisis anggur.

Metode

Adaboost (Adaptive Boosting) adalah algoritma ensemble learning yang dapat digunakan untuk meningkatkan kinerja model klasifikasi. Ensemble learning melibatkan penggabungan beberapa model untuk meningkatkan kinerja dan generalisasi model. Adaboost bekerja dengan memberikan "bobot" atau "bobot" yang berbeda pada setiap sampel data, memungkinkan model fokus pada sampel yang sulit diklasifikasikan.

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

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.

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.

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.

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.

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.

Mendapatkan Tingkat Kesalahan (error_rates Method)

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

Dalam kode tersebut, beberapa langkah dilakukan untuk menggunakan model Adaboost yang telah diimplementasikan sebelumnya. Berikut adalah penjelasan setiap langkah:

1. **Pelatihan Model Adaboost:**

ab = AdaBoost() ab.fit(X_train, y_train, M=400)

Sebuah objek **AdaBoost** dibuat, dan model tersebut dilatih menggunakan data pelatihan (**X_train** dan **y_train**). Model di-fit dengan 400 iterasi boosting (M = 400).

2. **Prediksi dan Evaluasi Model Terhadap Set Data Uji:**

from sklearn.metrics import roc_auc_score y_pred = ab.predict(X_test) print('Skor ROC-AUC dari model adalah:', round(roc_auc_score(y_test, y_pred), 4))

Model yang telah dilatih digunakan untuk melakukan prediksi pada set data uji (**X_test**), dan skor ROC-AUC dari model tersebut dihitung dan dicetak.

3. **Pembanding dengan Implementasi Scikit-learn:**

from sklearn.ensemble import AdaBoostClassifier ab_sk = AdaBoostClassifier(n_estimators=400) ab_sk.fit(X_train, y_train) y_pred_sk = ab_sk.predict(X_test) print('Skor ROC-AUC dari model adalah:', round(roc_auc_score(y_test, y_pred_sk), 4))

Dalam bagian ini, model Adaboost dari scikit-learn juga dilatih dengan parameter yang sama seperti model yang diimplementasikan. Skor ROC-AUC dari kedua model (model kustom dan model scikit-learn) dibandingkan untuk memvalidasi implementasi kustom.

4. **Visualisasi Tingkat Kesalahan Pelatihan:**

plt.figure(figsize=(10, 5)) plt.plot(ab.training_errors) plt.hlines(0.5, 0, 400, colors='red', linestyles='dashed') plt.title('Tingkat Kesalahan Pelatihan oleh Stump') plt.xlabel('Stump') plt.show()

Grafik digunakan untuk memvisualisasikan tingkat kesalahan selama pelatihan model (tingkat kesalahan untuk setiap iterasi boosting). Garis merah menunjukkan batas threshold (0.5).

5. **Visualisasi Tingkat Kesalahan Di Luar Sampel:**

ab.error_rates(X_test, y_test) plt.figure(figsize=(10, 5)) plt.plot(ab.prediction_errors) plt.hlines(0.5, 0, 400, colors='red', linestyles='dashed') plt.title('Tingkat Kesalahan Di Luar Sampel oleh Stump') plt.xlabel('Stump') plt.show()

Grafik ini menampilkan tingkat kesalahan di luar sampel (tingkat kesalahan untuk setiap iterasi pada data uji). Garis merah menunjukkan batas threshold (0.5).

6. **Perhitungan Tingkat Kesalahan Metaklasifier:**

print('Tingkat kesalahan dari metaklasifier:', round(hitung_kesalahan(y_test, y_pred, np.ones(len(y_test))), 4))

Tingkat kesalahan metaklasifier (model gabungan setelah boosting) dihitung menggunakan fungsi **hitung_kesalahan** dan dicetak.

Keseluruhan kode tersebut mencakup pelatihan model Adaboost, evaluasi model pada data uji, perbandingan dengan implementasi scikit-learn, dan visualisasi tingkat kesalahan.

Pengujian dan Hasil

Hasil pengujian metode Adaboost pada dataset yang diberikan menunjukkan hasil yang sangat baik:

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.
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.
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.
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

Dalam analisis data ini, 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.

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.

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

Kesimpulan dari analisis klasifikasi kelas anggur pada dataset tersebut adalah:

1. **Distribusi Fitur:**
    - Setiap kelas (Kelas_1, Kelas_2, dan Kelas_3) memiliki distribusi yang berbeda untuk setiap fitur pada dataset.
2. **Analisis Fitur:**
    - Melalui visualisasi distribusi dan nilai mean serta median, dapat ditemukan perbedaan karakteristik antar kelas untuk setiap fitur.
    - Sebagai contoh, fitur Alkohol, Asam Malat, Magnesium, Flavanoid, dan lainnya memiliki perbedaan yang signifikan antar kelas.

3. **Kesimpulan Analisis Fitur:**
    - Analisis distribusi fitur memberikan wawasan tentang bagaimana nilai-nilai fitur tersebut dapat membedakan antara kelompok-kelompok kelas.
    - Rata-rata dan median fitur-fitur tertentu dapat menjadi petunjuk penting untuk memahami perbedaan antar kelas anggur.
4. **Pemodelan dengan AdaBoost:**
    - Metode AdaBoost digunakan untuk meningkatkan kinerja model klasifikasi pada dataset wine.
    - Model yang dihasilkan menggunakan Decision Stump sebagai klasifier lemah.
5. **Evaluasi Model:**
    - Model dievaluasi menggunakan ROC-AUC Score, yang mencapai nilai maksimal (1.0), menunjukkan kinerja yang sangat baik.
    - Hasil model dari implementasi sendiri dibandingkan dengan implementasi dari library scikit-learn, dan konsistensi hasilnya diuji.
6. **Tingkat Kesalahan:**
    - Tingkat kesalahan selama pelatihan dan di luar sampel pada data uji dievaluasi, memberikan gambaran tentang kemampuan model untuk generalisasi.
7. **Kesimpulan Akhir:**
    - 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.

Dengan demikian, hasil analisis dan pemodelan ini memberikan pemahaman yang mendalam tentang karakteristik dataset kelas anggur dan keberhasilan metode AdaBoost dalam meningkatkan kinerja model klasifikasi.

Link video presentasi

https://drive.google.com/file/d/15_WeFJSKCs4ne4B2TWfZdJQ_Ho5CRPie/view?usp=sharing