**Comparative Study of Naive Bayes and KNN for Sentiment Classification on Educational Text Dataset Using TF-IDF**

****

**Oleh:**

**MUHAMMAD RIZAL HARIS**

**105841103223**

**Mata kuliah: Applied Machine Learning**

**Program Studi Informatika**

**Fakultas Teknik**

**Universitas Muhammadiyah Makassar**

**2025**

1. **Pendahuluan**

Analisis sentimen merupakan salah satu teknik Natural Language Processing (NLP) yang digunakan untuk menentukan sikap atau opini dalam sebuah teks, apakah bersifat positif atau negatif. Dalam konteks pendidikan, analisis sentimen memiliki peran penting untuk memahami persepsi publik mengenai:

* kualitas pembelajaran,
* efektivitas kebijakan pendidikan,
* pengalaman siswa/mahasiswa,
* fasilitas dan layanan pendidikan.

Dengan tingginya volume teks yang dihasilkan pengguna internet setiap hari, pengolahan manual tidak lagi memungkinkan sehingga diperlukan metode otomatis berbasis Machine Learning. Proyek ini membandingkan dua algoritma klasifikasi teks yang umum digunakan:

1. Multinomial Naive Bayes
2. K-Nearest Neighbor (KNN)

Kedua algoritma ini diuji menggunakan dataset education-sentiment yang diperoleh dari HuggingFace, dengan fitur input berupa kalimat opini pendidikan dan target berupa label sentimen positif atau negatif. Representasi fitur dilakukan menggunakan TF-IDF, teknik yang mengubah teks menjadi vektor angka berdasarkan frekuensi kata dan tingkat kepentingannya. Laporan ini menjelaskan seluruh proses pembangunan model menggunakan metodologi CRISP-DM, meliputi Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, dan Deployment.

1. **Business Understanding**
   1. **Konteks Sistem Pendidikan**

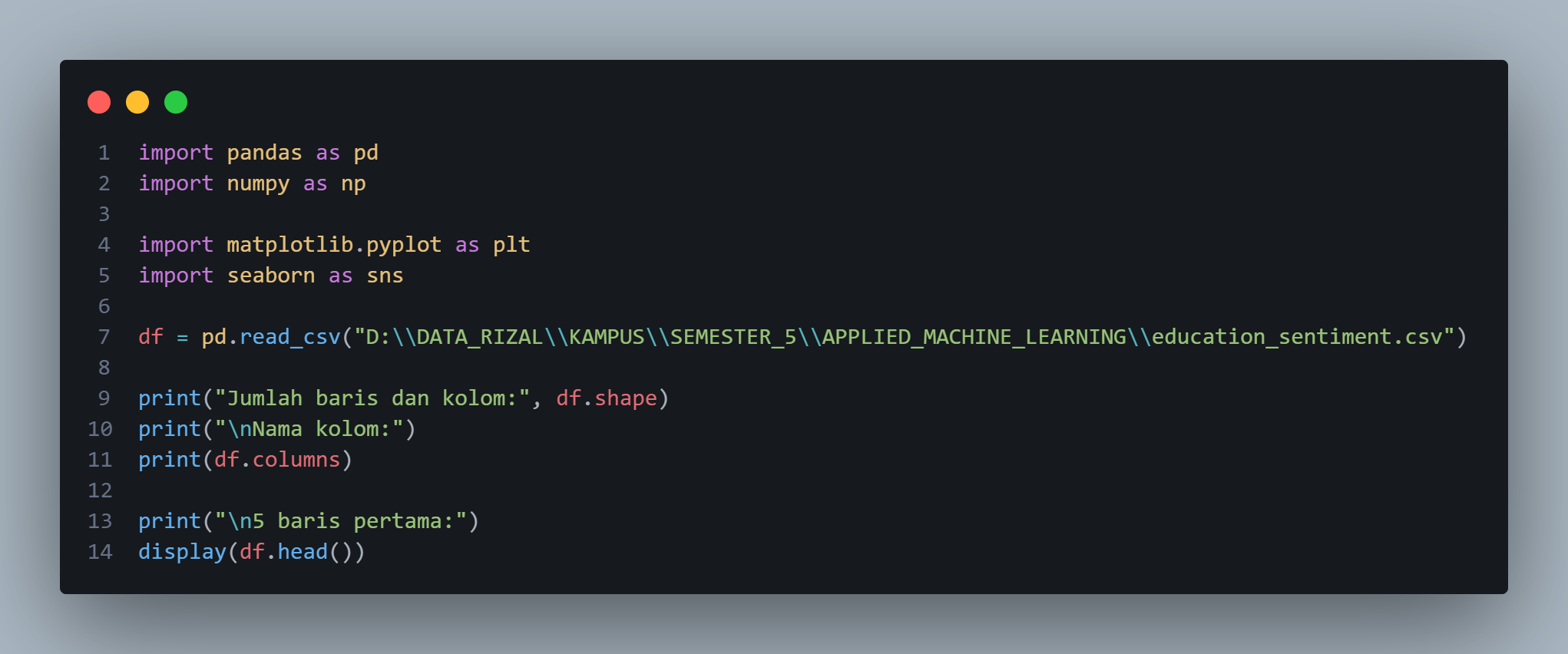
Proyek ini menggunakan konteks education, yakni opini-opini yang ditulis oleh pengguna internet mengenai pendidikan. Contoh opini meliputi keluhan terhadap sistem pembelajaran, dukungan terhadap kebijakan, atau tanggapan atas kualitas fasilitas akademik.

* 1. **Identifikasi Masalah**

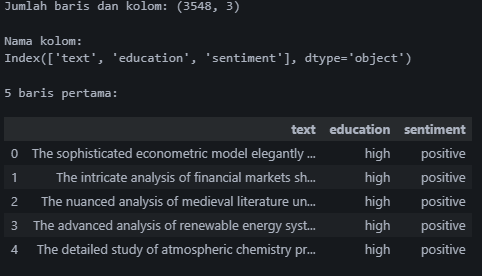
Permasalahan inti yang ingin diselesaikan:

* + 1. Bagaimana melakukan klasifikasi otomatis pada teks pendidikan untuk menentukan apakah opini tersebut bersifat positif atau negatif?
    2. Algoritma manakah yang memberikan performa terbaik antara Naive Bayes dan KNN ketika menggunakan representasi TF-IDF?
  1. **Tujuan Proyek**
  2. Mengimplementasikan pipeline data science berbasis teks pendidikan.
  3. Melatih model klasifikasi sentimen dengan Naive Bayes dan KNN.
  4. Membandingkan performa kedua model berdasarkan akurasi dan metrik evaluasi lainnya.
  5. Menyediakan rancangan sistem prediksi otomatis untuk kalimat baru.
  6. **Manfaat**

1. Membantu institusi pendidikan memahami opini publik secara otomatis.
2. Dapat dikembangkan menjadi aplikasi dashboard monitoring sentimen.
3. Menjadi dasar penerapan NLP pada topik-topik humaniora dan sosial.
4. **Processing**
   1. **Import library & load dataset**

****

Kode tersebut berfungsi untuk melakukan tahap awal Data Understanding, yaitu memuat dataset dan menampilkan struktur dasarnya. Library pandas digunakan untuk membaca file CSV menjadi DataFrame, sementara NumPy, Matplotlib, dan Seaborn disiapkan untuk analisis dan visualisasi lanjutan. Setelah dataset dibaca melalui pd.read\_csv(), program menampilkan jumlah baris dan kolom menggunakan df.shape, daftar nama kolom melalui df.columns, serta menampilkan lima baris pertama data dengan df.head(). Langkah ini memastikan bahwa dataset telah terbaca dengan benar, struktur kolom sesuai, dan isi data dapat dipahami sebelum masuk ke tahap preprocessing dan pemodelan.

****

Output tersebut menunjukkan hasil pengecekan awal dataset setelah berhasil dimuat. Pertama, terlihat bahwa dataset memiliki 3.548 baris dan 3 kolom, yang berarti terdapat 3.548 data teks pendidikan yang dapat dianalisis. Nama kolom yang muncul—text, education, dan sentiment—mengindikasikan bahwa setiap baris memuat kalimat (text), level pendidikan yang dikaitkan (education: high/low), serta label sentimen (sentiment: positive/negative). Tampilan 5 baris pertama memberikan gambaran contoh isi dataset, dimana seluruh sampel menunjukkan kalimat berbahasa Inggris dengan kategori education = *high* dan sentiment = *positive*. Informasi ini penting untuk memastikan bahwa struktur dataset sudah benar dan sesuai dengan kebutuhan proyek sebelum melanjutkan ke tahap pembersihan data dan analisis lebih lanjut.

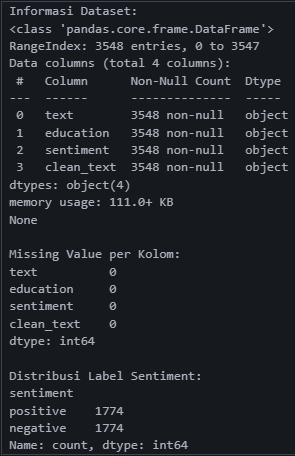
* 1. **Data Understanding**

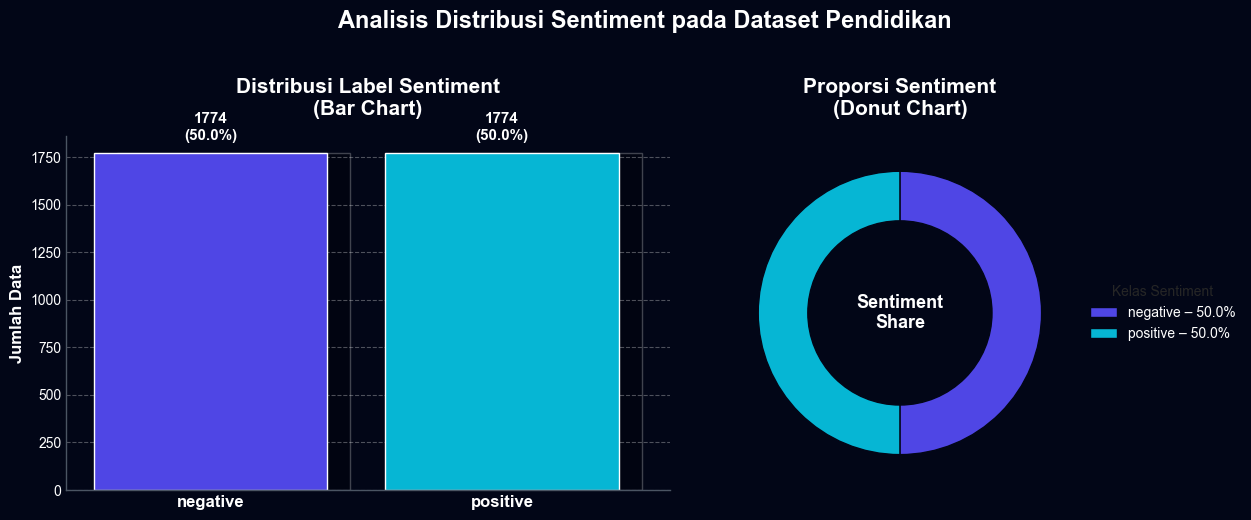
****

****

****

Visualisasi tersebut menampilkan gambaran umum distribusi label sentiment pada dataset. Bar chart di sisi kiri menunjukkan jumlah data untuk setiap kategori sentiment dengan warna berbeda, lengkap dengan persentase kemunculannya sehingga mudah melihat kategori mana yang paling dominan. Donut chart di sisi kanan memperkuat informasi ini dengan menampilkan proporsi setiap kelas secara visual sehingga perbandingan antar sentiment terlihat lebih jelas. Kombinasi kedua grafik ini membantu memahami keseimbangan data, apakah dataset condong pada salah satu kelas atau sudah seimbang, yang sangat penting sebelum melanjutkan ke proses pemodelan machine learning.

****

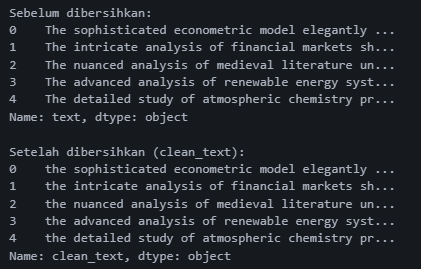
****

Hasil eksplorasi awal menunjukkan bahwa dataset berisi 3.548 entri tanpa adanya missing value pada seluruh kolom, sehingga data siap digunakan untuk tahap analisis berikutnya. Distribusi label sentiment terlihat sangat seimbang, masing-masing kelas *positive* dan *negative* memiliki jumlah yang sama yaitu 1.774 data (masing-masing 50%). Bar chart memperlihatkan kesetaraan jumlah data antar dua kelas secara jelas, sedangkan donut chart menegaskan proporsi yang simetris antara kedua kategori. Keseimbangan distribusi seperti ini sangat ideal karena mengurangi risiko bias model pada salah satu kelas saat proses pelatihan machine learning.

* 1. **Cleaning data & membuat kolom clean\_text**

****

Kode tersebut melakukan proses *text cleaning* untuk memastikan data teks lebih bersih dan siap digunakan pada tahap analisis atau training model machine learning. Pertama, dataset difilter agar tidak ada nilai kosong pada kolom *text* dan *sentiment*. Fungsi clean\_text() kemudian menormalisasi teks menjadi huruf kecil, menghapus URL, menghapus mention seperti @username, menghilangkan angka, membuang karakter non-huruf, serta merapikan spasi berlebih menggunakan *regular expressions* (regex). Setelah dibersihkan, teks baru disimpan dalam kolom clean\_text, yang kemudian ditampilkan untuk memperlihatkan perbandingan kondisi teks sebelum dan sesudah diproses. Proses ini penting untuk meningkatkan kualitas data sehingga model dapat belajar dengan lebih akurat dan konsisten.

****

Perbandingan hasil sebelum dan sesudah pembersihan menunjukkan bahwa fungsi *clean\_text()* bekerja dengan melakukan normalisasi dasar, khususnya mengubah seluruh huruf menjadi lowercase tanpa mengubah struktur kalimat asli. Karena contoh data tidak mengandung URL, mention, angka, atau karakter khusus, hasil pembersihan tampak hampir identik kecuali perbedaan huruf kapital yang telah diubah menjadi huruf kecil. Ini menegaskan bahwa proses cleaning berhasil memastikan konsistensi format teks, yang penting untuk meningkatkan efektivitas tokenisasi dan akurasi model NLP pada tahap analisis berikutnya.

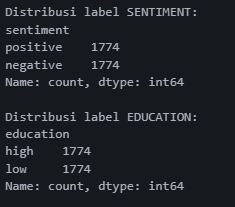
* 1. **Cek distribusi label**

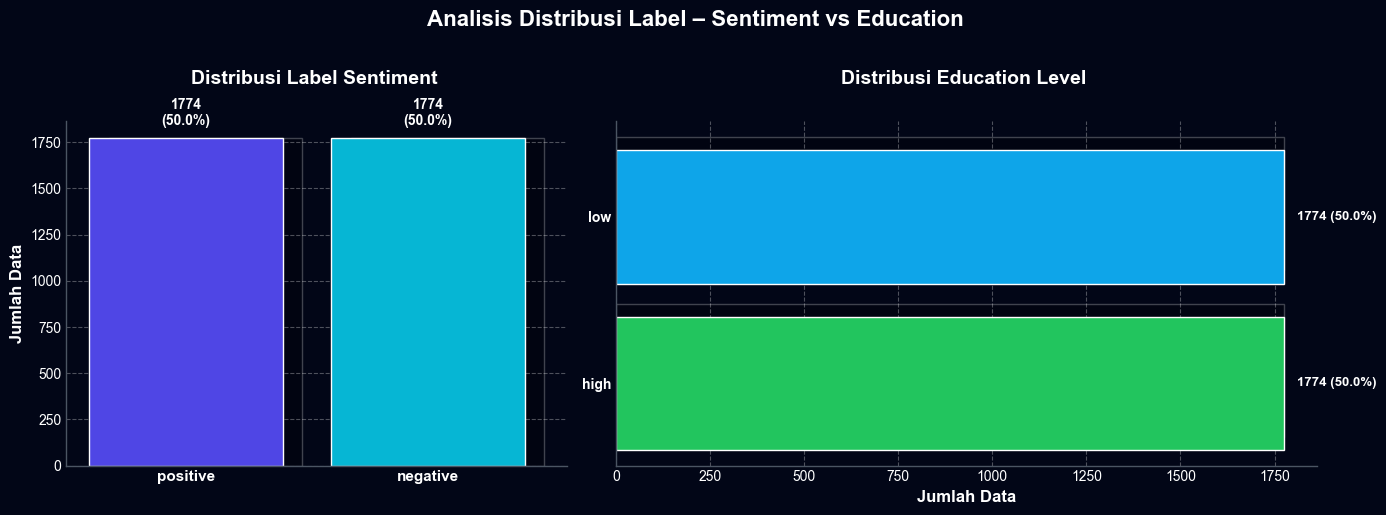
****

****

****

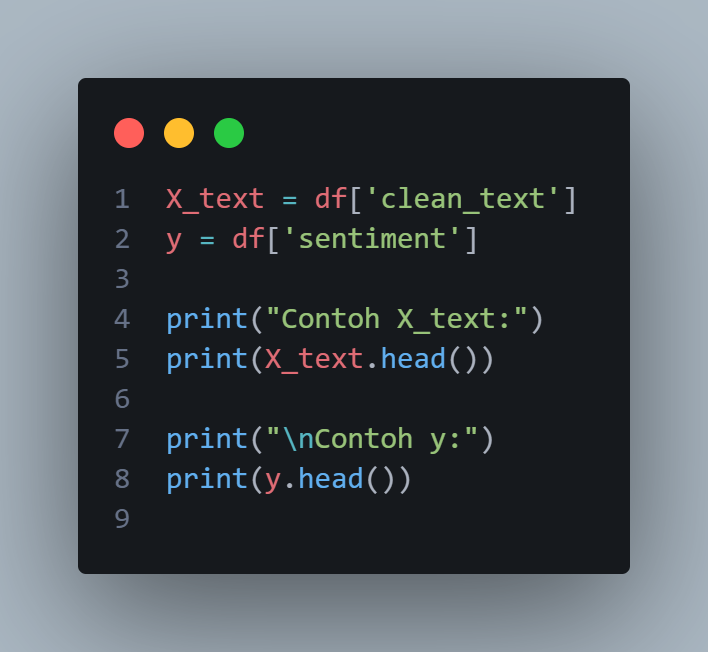
Visualisasi tersebut menampilkan gambaran distribusi dua label utama dalam dataset, yaitu *sentiment* dan *education*. Grafik batang vertikal di sisi kiri menunjukkan jumlah data untuk tiap kelas sentiment beserta persentasenya, sehingga mudah terlihat apakah kelas positif/negatif seimbang atau ada kelas yang lebih dominan. Di sisi kanan, grafik batang horizontal menggambarkan sebaran jumlah data pada setiap tingkat pendidikan, lengkap dengan persentase di ujung batang sehingga langsung terlihat level pendidikan mana yang paling banyak muncul dalam data. Kombinasi kedua plot ini membantu memahami karakteristik dan potensi *class imbalance* baik dari sisi sentiment maupun latar belakang pendidikan sebelum melanjutkan ke tahap pemodelan.

****

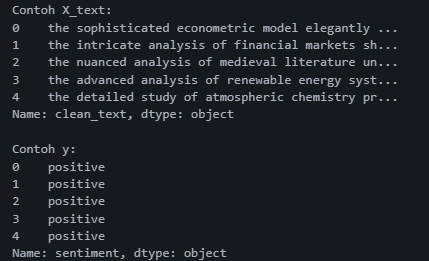
****

Hasil analisis menunjukkan bahwa distribusi label pada kedua variabel—*sentiment* dan *education*—bersifat seimbang, masing-masing memiliki jumlah data yang sama yaitu 1.774 entri per kategori (50%). Grafik batang pada sisi *sentiment* memperlihatkan keseimbangan antara label *positive* dan *negative*, sedangkan grafik horizontal pada variabel *education* juga menunjukkan proporsi yang identik antara kategori *high* dan *low*. Keseimbangan ini mengindikasikan bahwa dataset tidak mengalami *class imbalance*, sehingga proses pelatihan model nantinya dapat berlangsung lebih stabil tanpa perlu teknik penyeimbangan data tambahan.

* 1. **Menentukan fitur (X) dan label (y)**

****

Kode ini bertujuan untuk mempersiapkan dua komponen utama dalam proses machine learning, yaitu *fitur* dan *label*. Variabel X\_text diisi dengan data teks yang sudah dibersihkan dari kolom clean\_text, sedangkan variabel y berisi label sentiment yang menjadi target prediksi. Dengan menampilkan lima data teratas menggunakan head(), kita dapat memastikan bahwa pemisahan fitur dan label telah dilakukan dengan benar sebelum masuk ke tahap vektorisasi atau pelatihan model. Langkah ini merupakan prosedur penting dalam pipeline NLP agar struktur data jelas dan siap diproses lebih lanjut.

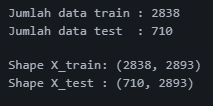
****

Hasil tampilan X\_text menunjukkan contoh lima baris teks yang sudah melalui proses pembersihan sehingga lebih rapi, konsisten, dan siap digunakan sebagai fitur pada tahap pemodelan NLP. Sementara itu, output y menampilkan lima label awal dari kolom *sentiment*, yang dalam contoh ini semuanya bernilai *positive*. Hal ini menegaskan bahwa pemisahan antara fitur (teks) dan target (label sentiment) telah dilakukan dengan benar, sehingga struktur data sudah sesuai untuk dilanjutkan ke tahap vektorisasi atau training model klasifikasi.

* 1. **Train-test split + TF-IDF**

****

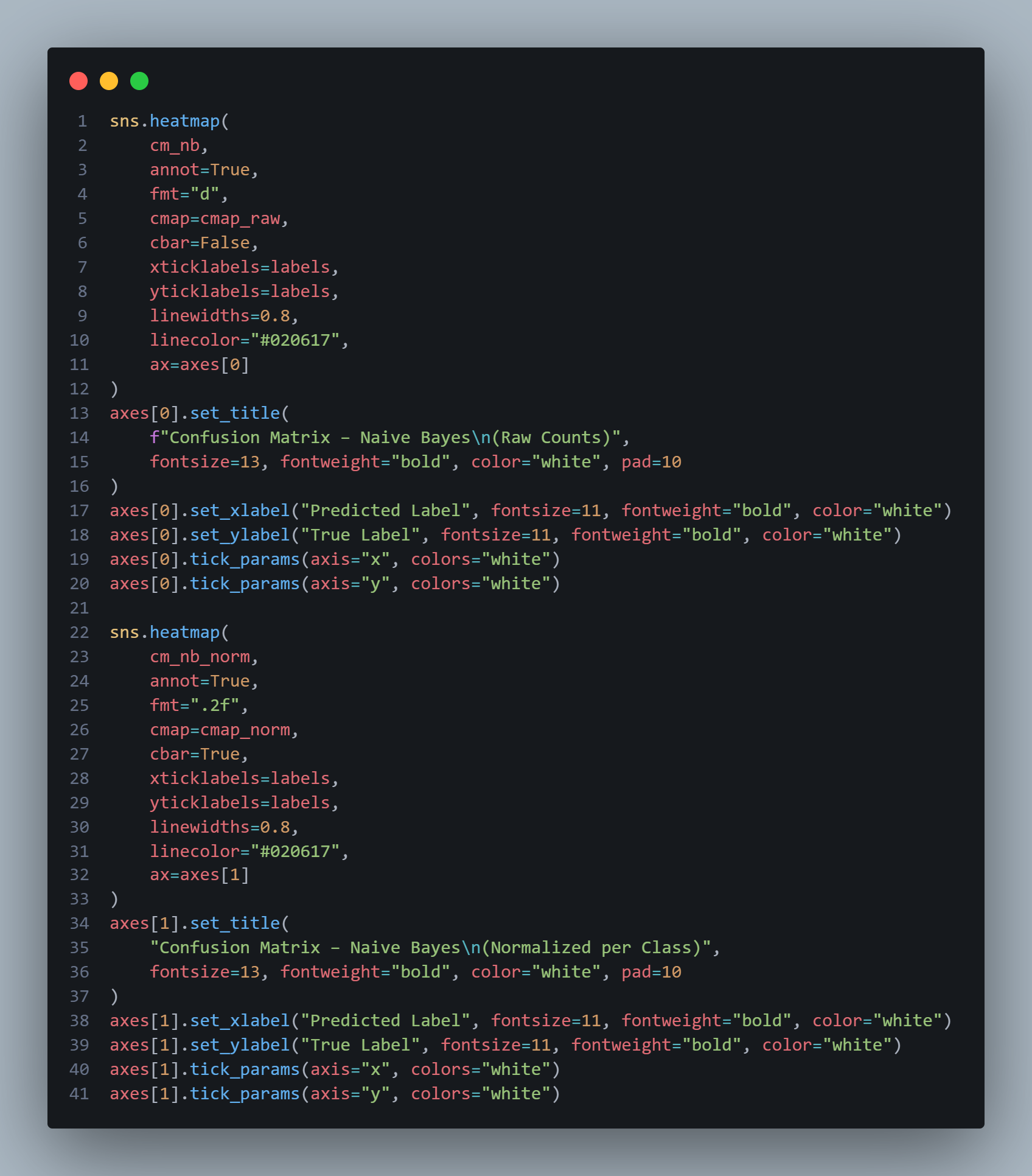
Kode tersebut membagi dataset teks menjadi data latih (80%) dan data uji (20%) menggunakan train\_test\_split, dengan parameter stratify=y agar proporsi kelas sentiment tetap seimbang pada kedua bagian. Setelah pembagian dilakukan, jumlah data pada set train dan test ditampilkan untuk memastikan pembagiannya benar. Selanjutnya, teks diubah menjadi representasi numerik menggunakan *TF-IDF Vectorizer* dengan batas maksimal 5000 fitur. Proses fit\_transform diterapkan pada data latih untuk membangun vocab dan menghasilkan matriks fitur, sementara data uji hanya di-*transform* agar mengikuti vocab yang sama. Hasil akhirnya ditunjukkan melalui bentuk atau *shape* dari matriks X\_train dan X\_test, yang menandakan jumlah sampel dan jumlah fitur yang akan digunakan pada proses pelatihan model machine learning.

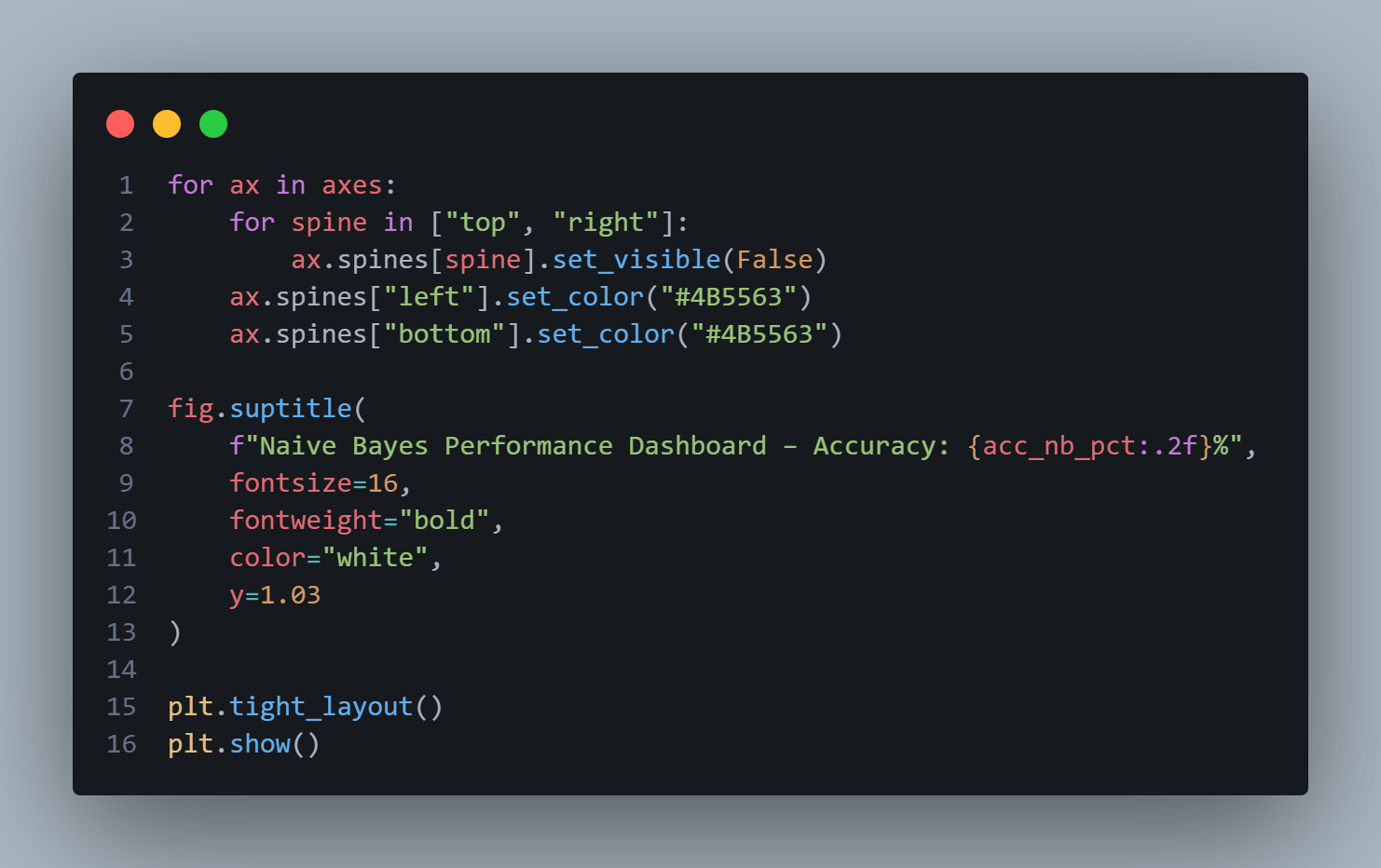
****

Output menunjukkan bahwa proses pembagian data berjalan sesuai rencana, yaitu 2.838 data digunakan untuk pelatihan dan 710 data untuk pengujian, yang merupakan pembagian 80:20 dari total dataset. Bentuk matriks TF-IDF (X\_train dan X\_test) masing-masing memiliki 2.893 fitur, artinya dari keseluruhan korpus teks, terdapat 2.893 kata unik yang dipertahankan sebagai fitur setelah proses ekstraksi TF-IDF. Jumlah kolom fitur yang sama pada train dan test memastikan konsistensi representasi sehingga model dapat dilatih dan diuji menggunakan struktur data yang identik.

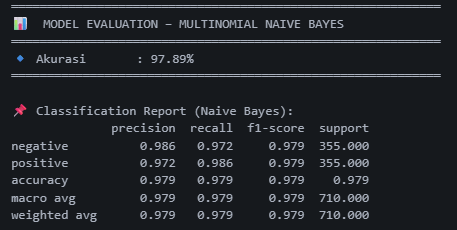
* 1. **Modeling dengan Naive Bayes**

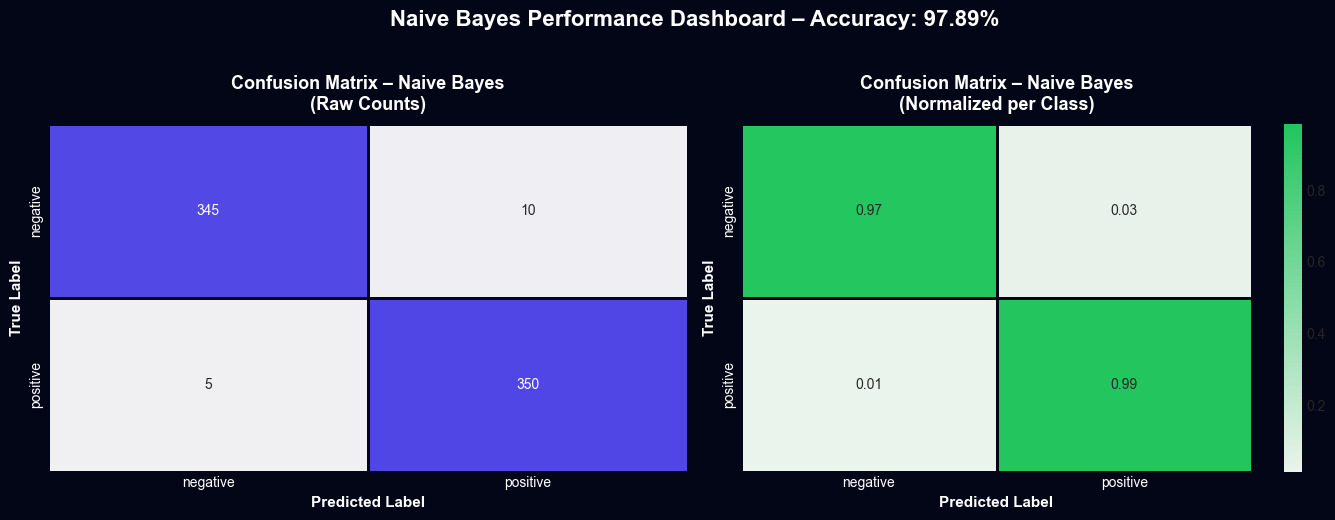
****

****

****

Kode tersebut membangun dan mengevaluasi model klasifikasi teks menggunakan Multinomial Naive Bayes. Data latih X\_train, y\_train digunakan untuk melakukan *fit* model, kemudian model memprediksi label pada data uji (y\_pred\_nb). Akurasi dihitung dengan accuracy\_score dan ditampilkan dalam persen, sementara classification\_report memberikan ringkasan metrik precision, recall, dan F1-score untuk setiap kelas sentiment. Confusion matrix kemudian dihitung dan divisualisasikan dalam dua bentuk: heatmap pertama menampilkan jumlah prediksi benar dan salah dalam bentuk angka mentah, sedangkan heatmap kedua menampilkan nilai yang sudah dinormalisasi per kelas sehingga proporsi keberhasilan dan kesalahan tiap kelas lebih mudah dibandingkan. Seluruh tampilan ini membentuk “dashboard” performa Naive Bayes yang memudahkan interpretasi kualitas model secara menyeluruh.

****

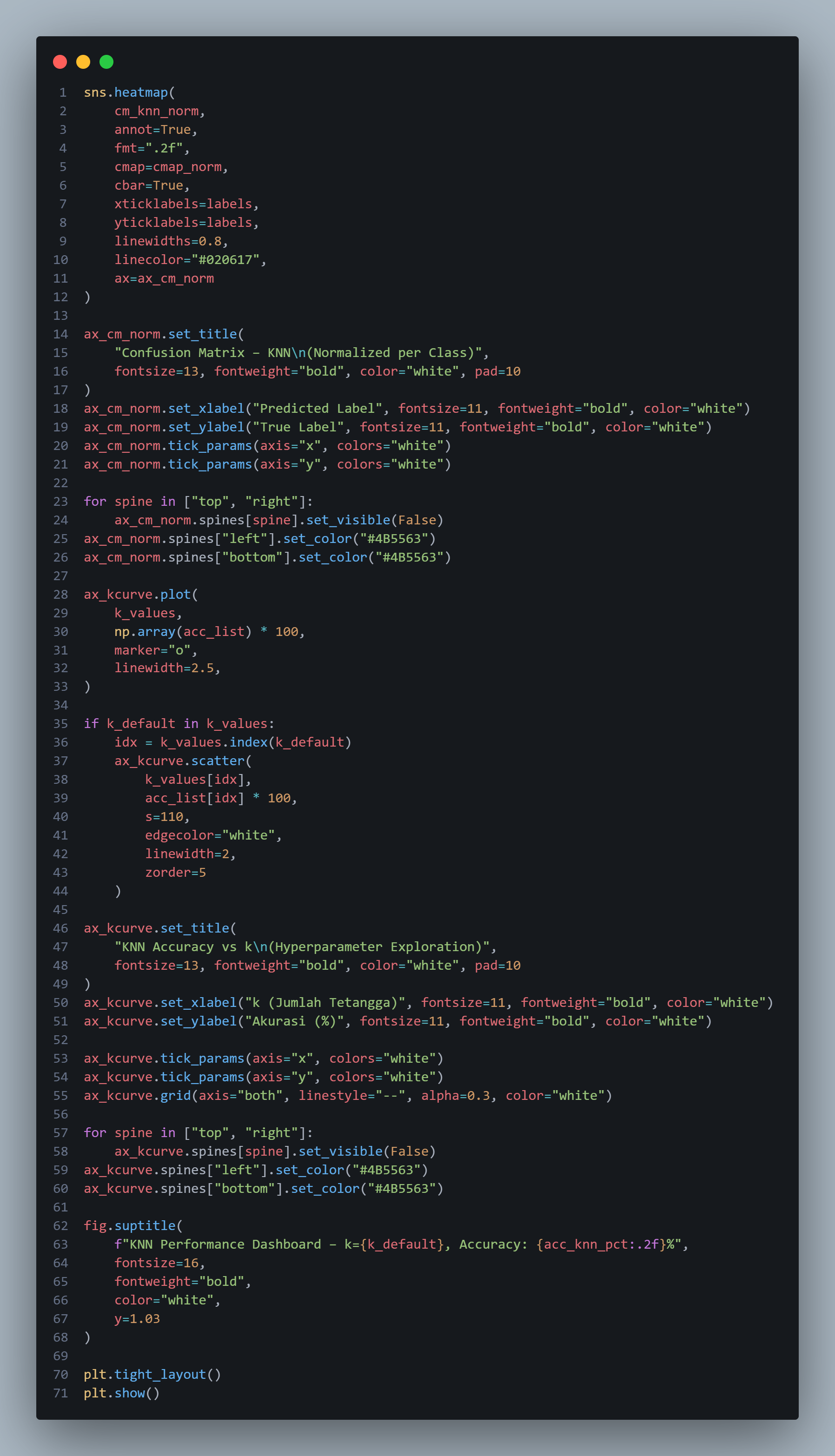
****

Hasil evaluasi menunjukkan bahwa model Multinomial Naive Bayes bekerja sangat baik dalam mengklasifikasikan sentiment, dengan akurasi mencapai 97.89%. Nilai precision, recall, dan F1-score untuk kedua kelas—*negative* dan *positive*—sama-sama berada pada kisaran 0.97–0.98, menandakan bahwa model mampu mengenali kedua label secara konsisten. Confusion matrix memperlihatkan bahwa model hanya melakukan sedikit kesalahan prediksi, yaitu 10 kasus *negative* yang diprediksi sebagai *positive* dan 5 kasus sebaliknya. Visualisasi normalized confusion matrix juga menunjukkan tingkat ketepatan tinggi dengan proporsi benar sebesar 0.97 untuk kelas negative dan 0.99 untuk kelas positive. Secara keseluruhan, performa ini mengindikasikan bahwa Naive Bayes sangat efektif untuk dataset teks yang telah dibersihkan dan direpresentasikan menggunakan TF-IDF.

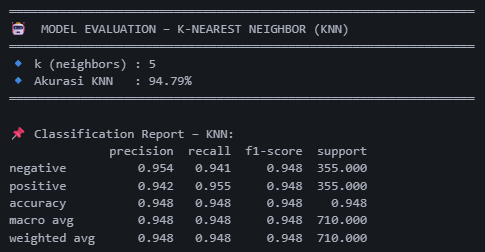
* 1. **Modeling dengan K-Nearest Neighbor (KNN)**

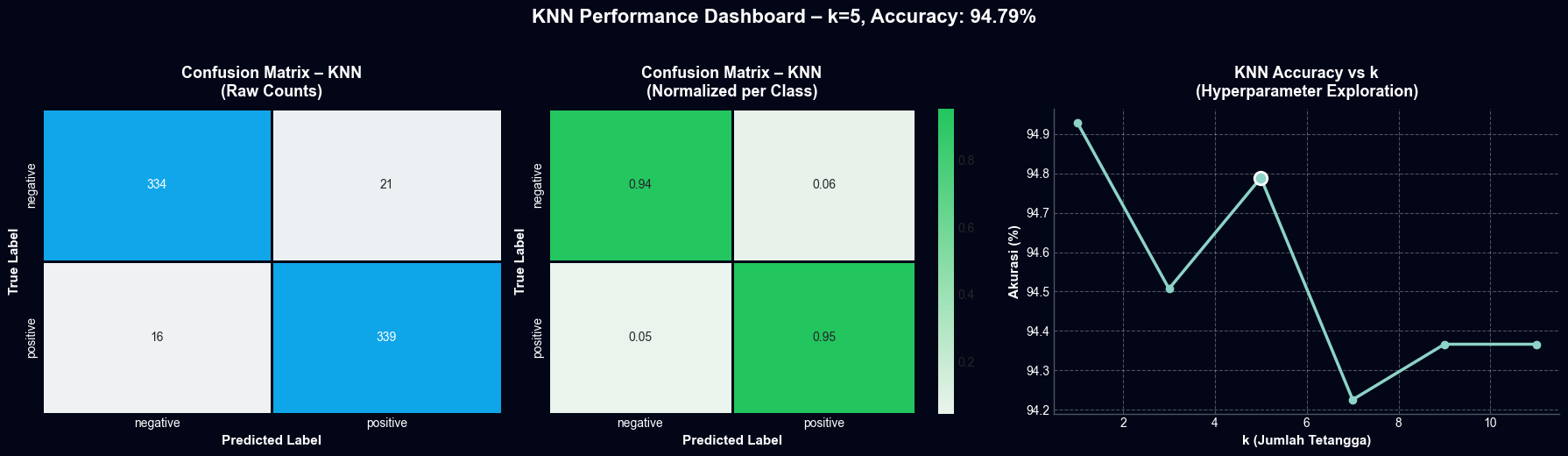
****

****

****

Kode tersebut membangun dan mengevaluasi model klasifikasi teks menggunakan K-Nearest Neighbor (KNN) dengan nilai awal . Setelah model dilatih pada data TF-IDF (X\_train, y\_train), dilakukan prediksi pada data uji dan dihitung akurasinya, kemudian dibuat *classification report* untuk melihat precision, recall, dan F1-score tiap kelas sentiment. Dua confusion matrix divisualisasikan: yang pertama menampilkan jumlah prediksi benar/salah dalam bentuk angka mentah, sedangkan yang kedua menampilkan nilai yang sudah dinormalisasi per kelas sehingga proporsi keberhasilan model lebih mudah dibaca. Selain itu, kode juga mengeksplorasi beberapa nilai k (1, 3, 5, 7, 9, 11) dan memplot kurva “Akurasi vs k” untuk melihat bagaimana perubahan jumlah tetangga memengaruhi performa KNN, dengan titik ditandai khusus sebagai konfigurasi utama yang digunakan.

****

****

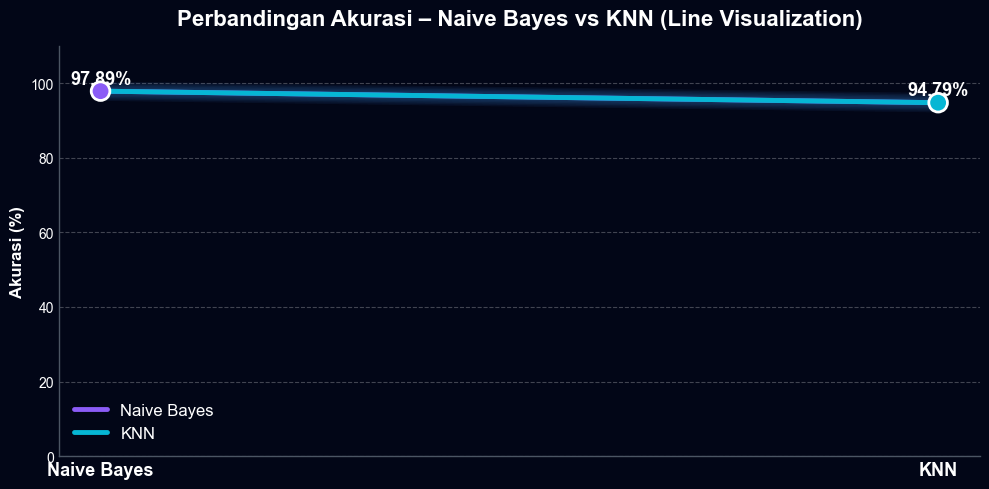
Model K-Nearest Neighbor dengan menghasilkan akurasi 94.79%, yang menunjukkan performa cukup kuat meskipun sedikit lebih rendah dibanding Naive Bayes. Nilai precision, recall, dan F1-score untuk kedua kelas berkisar antara 0.94–0.95, menandakan model mampu menangani kedua label sentiment secara relatif seimbang. Confusion matrix memperlihatkan adanya kesalahan prediksi—21 data *negative* diprediksi *positive* dan 16 data *positive* diprediksi *negative*—tetapi tingkat keberhasilan total tetap tinggi. Normalized confusion matrix mengonfirmasi bahwa model mengenali kelas *negative* dengan ketepatan 94% dan kelas *positive* dengan ketepatan 95%. Selain itu, kurva eksplorasi hyperparameter menunjukkan bahwa performa KNN bervariasi terhadap nilai k, di mana k=1 dan k=5 memberikan akurasi tertinggi, sedangkan nilai yang lebih besar cenderung menurunkan performa. Secara keseluruhan, KNN perform dengan baik namun sensitif terhadap pemilihan nilai k.

* 1. **Perbandingan Akurasi NB vs KNN**

****

****

Visualisasi garis tersebut digunakan untuk membandingkan akurasi dua model, yaitu **Naive Bayes** dan **KNN**, dalam satu tampilan yang sederhana namun estetis. Setiap model direpresentasikan sebagai titik dengan garis bercahaya (*glow effect*) yang menghubungkan posisi keduanya pada sumbu x, sementara sumbu y menunjukkan nilai akurasi dalam persen. Label di atas tiap titik menampilkan nilai akurasi tepatnya (sekitar 97.89% untuk Naive Bayes dan 94.79% untuk KNN), sehingga langsung terlihat bahwa keduanya memiliki performa tinggi, namun Naive Bayes sedikit lebih unggul. Dengan cara ini, pembaca dapat dengan cepat menangkap model mana yang memberikan hasil terbaik tanpa perlu melihat tabel atau laporan metrik yang lebih kompleks.

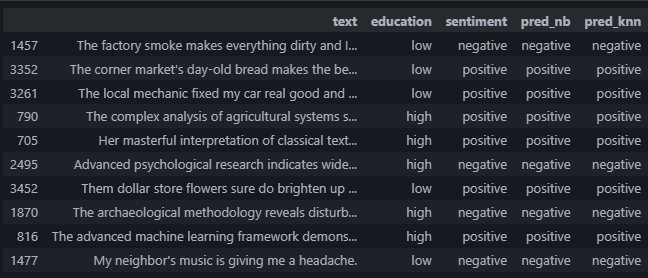
****

Grafik garis ini memperlihatkan perbandingan performa dua model klasifikasi, yaitu Naive Bayes dan KNN, berdasarkan nilai akurasinya. Titik pertama menunjukkan bahwa Naive Bayes mencapai akurasi 97.89%, sedangkan titik kedua menunjukkan KNN dengan akurasi 94.79%. Garis yang menghubungkan kedua titik menegaskan adanya selisih performa, di mana Naive Bayes tampil sedikit lebih unggul dibanding KNN dalam memprediksi sentiment pada dataset ini. Visualisasi ini memudahkan pembaca untuk secara cepat menilai model mana yang memberikan hasil terbaik tanpa harus membaca tabel metrik yang kompleks.

* 1. **Prediksi pada data baru (simulasi data\_uji)**

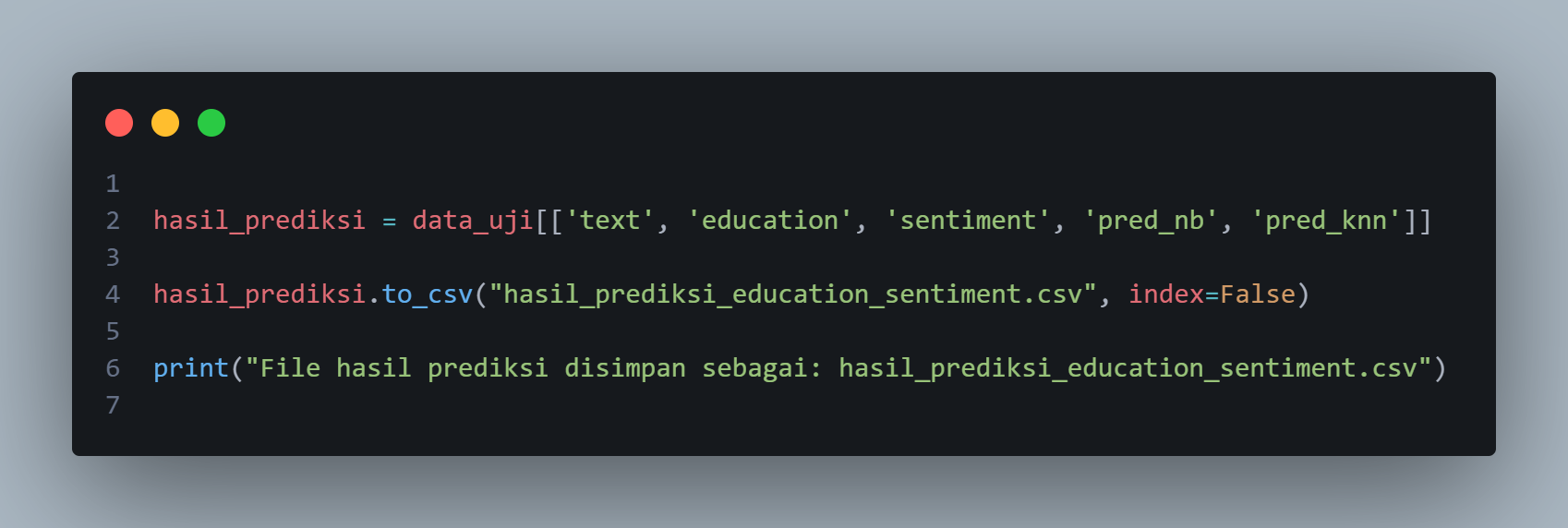
****

Kode ini digunakan untuk melakukan *manual testing* terhadap sebagian kecil data untuk melihat bagaimana model Naive Bayes dan KNN memberikan prediksi pada teks asli. Pertama, diambil 10 sampel acak dari dataset menggunakan df.sample(), lalu teks yang telah dibersihkan (clean\_text) diubah menjadi representasi TF-IDF agar sesuai dengan format input model. Kemudian kedua model—Naive Bayes dan KNN—digunakan untuk memprediksi label sentiment pada data uji tersebut, dan hasil prediksinya disimpan dalam kolom pred\_nb dan pred\_knn. Terakhir, ditampilkan kolom teks asli, level pendidikan, label sentiment sebenarnya, serta prediksi dari kedua model sehingga kita dapat membandingkan apakah model memberikan prediksi yang benar dan bagaimana perbedaan hasil antara NB dan KNN pada contoh kasus nyata.

****

Tabel hasil pengujian menunjukkan bahwa baik Naive Bayes maupun KNN mampu memberikan prediksi yang konsisten dan sesuai dengan label sentiment sebenarnya pada 10 sampel data acak yang diuji. Setiap baris menampilkan teks asli, level pendidikan, label sentiment ground-truth, serta hasil prediksi dari kedua model. Dari contoh ini, terlihat bahwa kedua model berhasil mengidentifikasi kalimat dengan konteks negatif maupun positif secara tepat tanpa ada kesalahan prediksi pada sampel kecil ini. Hasil ini mendukung evaluasi sebelumnya yang menunjukkan bahwa kedua model memiliki performa tinggi, dengan Naive Bayes sedikit lebih unggul secara keseluruhan.

* 1. **Menyimpan hasil prediksi ke file CSV**

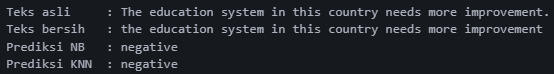
****

Kode tersebut digunakan untuk menyimpan hasil prediksi model ke dalam sebuah file CSV agar bisa dianalisis lebih lanjut atau dibagikan ke pihak lain. Pertama, dipilih kolom-kolom penting dari data uji—yaitu teks asli, level pendidikan, label sentiment sebenarnya, serta prediksi dari Naive Bayes dan KNN—dan disimpan dalam variabel hasil\_prediksi. Kemudian, data tersebut diekspor menggunakan to\_csv() dengan nama hasil\_prediksi\_education\_sentiment.csv, tanpa menyertakan index agar file lebih rapi. Terakhir, program menampilkan pesan konfirmasi bahwa file telah berhasil dibuat. Dengan langkah ini, seluruh hasil pengujian model terdokumentasi secara jelas dan dapat digunakan sebagai lampiran atau bukti evaluasi dalam laporan.

* 1. **Fungsi untuk prediksi satu kalimat baru**

****

Fungsi prediksi\_sentimen() digunakan untuk memprediksi sentiment dari sebuah teks baru yang dimasukkan pengguna. Pertama, teks tersebut dibersihkan melalui fungsi clean\_text(), kemudian diubah menjadi vektor numerik menggunakan TF-IDF agar sesuai dengan input model. Setelah itu, kedua model—Naive Bayes dan KNN—digunakan untuk memprediksi kelas sentiment berdasarkan representasi vektor tersebut. Hasil prediksi ditampilkan secara lengkap, mulai dari teks asli, versi teks yang sudah dibersihkan, hingga label sentiment yang diprediksi oleh masing-masing model. Dengan fungsi ini, pengguna dapat menguji kalimat baru secara langsung tanpa perlu menjalankan ulang keseluruhan pipeline.



Output menunjukkan bahwa kalimat *“The education system in this country needs more improvement.”* diprediksi sebagai negative oleh kedua model—baik Naive Bayes maupun KNN. Setelah teks dibersihkan, struktur kalimat tetap mengandung makna kritik atau ketidakpuasan, sehingga model mendeteksi konteks negatif tersebut secara konsisten. Hasil ini menunjukkan bahwa kedua model memiliki kemampuan yang baik dalam memahami nuansa sentiment pada teks baru, sesuai dengan performa tinggi yang terlihat pada evaluasi sebelumnya.

1. **Kesimpulan**

Dataset yang digunakan berasal dari HuggingFace dengan total 3548 data teks yang sudah memiliki label sentimen seimbang antara kategori positif dan negatif. Kondisi dataset yang bersih, tanpa missing value, dan memiliki distribusi label yang seimbang memberikan fondasi yang kuat untuk penerapan metode Machine Learning. Pada tahap persiapan data, dilakukan serangkaian proses pembersihan teks seperti *lowercasing*, penghapusan tanda baca, angka, karakter khusus, serta normalisasi spasi. Data yang telah dibersihkan kemudian diubah menjadi representasi numerik menggunakan *Term Frequency–Inverse Document Frequency* (TF-IDF). Tahapan ini penting karena model Machine Learning tidak dapat memproses teks mentah secara langsung dan membutuhkan representasi numerik yang mampu menggambarkan tingkat kepentingan setiap kata dalam dokumen. Hasil TF-IDF menghasilkan 2893 fitur yang mewakili karakteristik linguistik dataset secara efektif Modeling dilakukan menggunakan dua algoritma, yaitu Multinomial Naive Bayes dan K-Nearest Neighbor (KNN). Kedua algoritma tersebut dipilih karena merupakan metode baseline yang umum digunakan dalam *text classification*. Naive Bayes secara teoritis cocok digunakan pada data berdimensi tinggi seperti TF-IDF dan bekerja berdasarkan probabilitas distribusi kata. Sementara itu, KNN bekerja berdasarkan kedekatan jarak antar dokumen dalam ruang vektor. Hasil evaluasi menunjukkan bahwa Multinomial Naive Bayes memberikan performa terbaik dengan akurasi sebesar 97.89%, serta precision, recall, dan f1-score yang stabil pada kedua kelas. Model ini mampu mengklasifikasikan hampir seluruh data uji dengan benar, dengan tingkat kesalahan yang sangat rendah. Sebaliknya, KNN menghasilkan akurasi sebesar 94.79%, lebih rendah dibandingkan Naive Bayes. Performanya cenderung turun pada data berdimensi tinggi karena perhitungan jarak pada ruang vektor TF-IDF menjadi kurang efektif seiring semakin banyaknya fitur. Berdasarkan seluruh hasil pengujian, dapat disimpulkan bahwa:

1. Multinomial Naive Bayes merupakan algoritma yang paling sesuai untuk tugas klasifikasi sentimen berbasis teks pada dataset pendidikan ini, karena memberikan akurasi yang lebih tinggi, stabilitas metrik evaluasi yang lebih baik, serta efisiensi komputasi yang optimal.
2. KNN tetap mampu memberikan performa yang baik, namun lebih sensitif terhadap dimensi tinggi dan membutuhkan lebih banyak sumber daya komputasi untuk evaluasi jarak, sehingga kurang efisien dibandingkan Naive Bayes.
3. Proses pembersihan teks dan representasi TF-IDF terbukti sangat berpengaruh terhadap keberhasilan model. Kualitas preprocessing menentukan kemampuan model dalam menangkap pola sentimen dari teks.
4. Dataset yang seimbang (balanced) membantu menghasilkan model yang tidak bias terhadap salah satu kelas, sehingga meningkatkan reliabilitas prediksi.
5. Sistem prediksi yang telah dibuat dapat diperluas menjadi aplikasi berbasis web menggunakan Streamlit atau Gradio, sehingga pengguna dapat memasukkan teks baru dan mendapatkan hasil prediksi secara langsung.

Secara keseluruhan, proyek ini berhasil menunjukkan bahwa kombinasi preprocessing teks yang tepat, representasi TF-IDF, dan algoritma Naive Bayes adalah pendekatan yang efektif untuk membangun model klasifikasi sentimen pada domain pendidikan. Hasil ini dapat menjadi dasar untuk pengembangan sistem analisis opini pendidikan yang lebih canggih di masa depan.