**Laporan Kuis 1**

**Mata kuliah: Machine Learning**

Dosen Pengampu Mata Kuliah: Irsyad Arif Mashudi, S.Kom., M.Kom.



Disusun oleh:

Zhubair Abhel

(2141720248 / 30)

Kelompok 6 Kelas TI-3D

**JURUSAN TEKNOLOGI INFORMASI PROGRAM STUDI D4 TEKNIK INFORMATIKA POLITEKNIK NEGERI MALANG**

**TAHUN 2023**

Kasus - Klasifikasi Tulisan Tangan dengan Dataset MNIST Deskripsi Umum

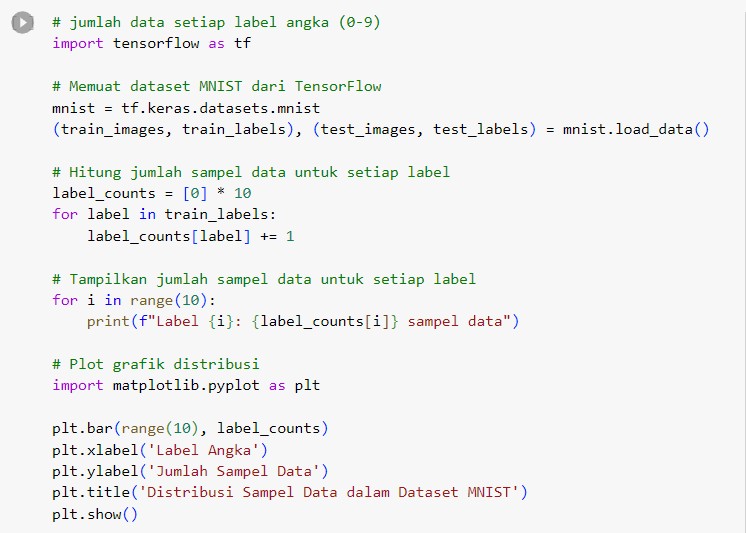
Anda diminta untuk melakukan klasifikasi dengan menggunakan algoritma Naive Bayes dan SVMuntuk merekognisi tulisan tangan dari dataset MNIST

Tentang Dataset MNIST

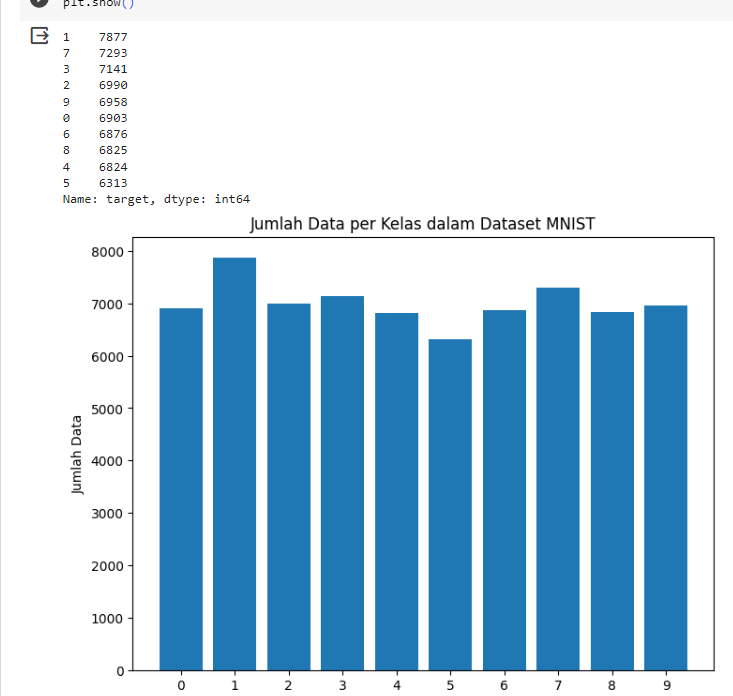
Dataset MNIST (Modified National Institute of Standards and Technology) merupakan dataset berupacitra grayscale dengan ukuran 28x28 yang berisi tulisan tangan dari digit angka 0-9. Jumlah datadalam dataset ini adalah 70.000 data.

1. Pastikan jumlah data setiap label angka (0-9). Apakah terdapat data imbalance pada dataset MNIST yang digunakan? Jelaskan jawaban kelompok Anda dan buktikan!

Kode program :



Hasil running:

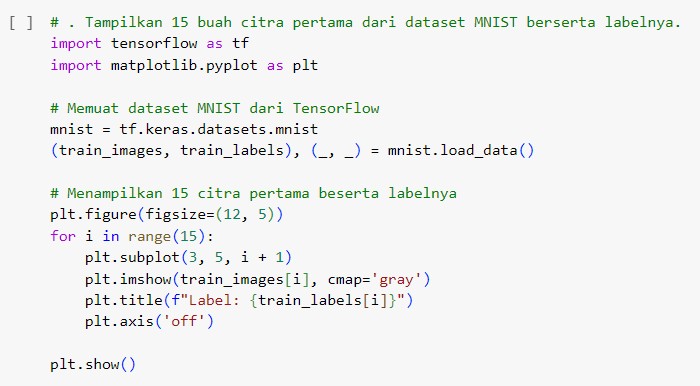


# Penjelasan:

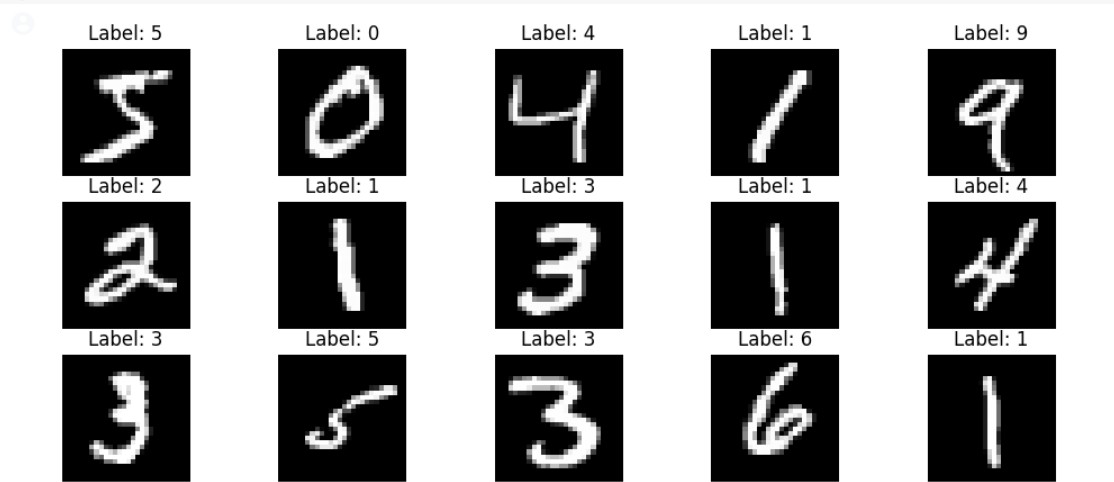
Dataset MNIST dianggap seimbang karena distribusi jumlah sampel pada setiap label (0-9) hampir sama baik dalam set pelatihan maupun set pengujian. Hal ini menggambarkan bahwa dataset dianggap seimbang dalam konteks klasifikasi. Ketidakseimbangan kelas dapat menyebabkan masalah, seperti model pembelajaran mesin cenderung memprediksi kelas mayoritas dan mengabaikan kelas minoritas, yang dapat mengurangi performa model terutama pada kelas minoritas. Oleh karena itu, dataset yang seimbang memungkinkan model untuk belajar dengan baik dari semua kelas, dan ini dapat meningkatkan performa keseluruhan model dalam tugas klasifikasi.

1. Tampilkan 15 buah citra pertama dari dataset MNIST berserta labelnya. Anda dapat menggunakan referensi dari Job Sheet 03.

Kode program :



* + Hasil running :



# Penjelasan:

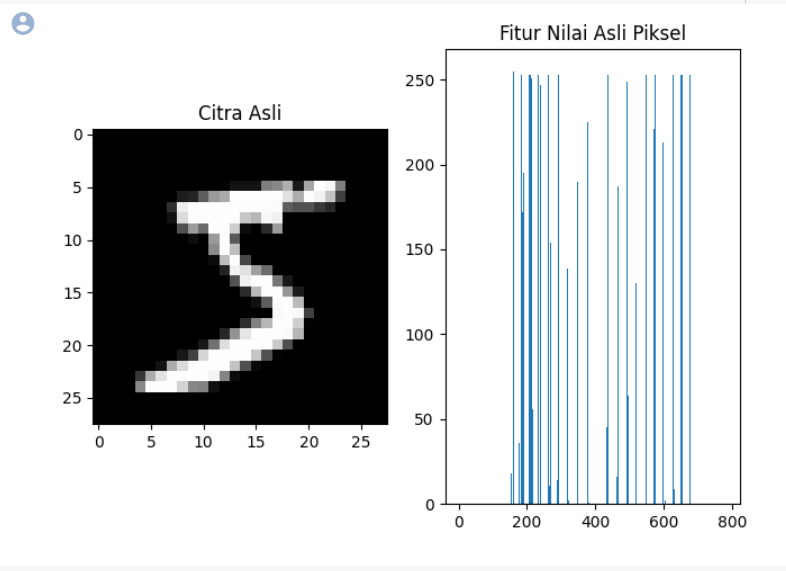
Menggunakan pustaka tensorflow dan matplotlib yang digunakan untuk memuat dataset MNIST dan kemudian menampilkannya. Kemudian menggunakan perulangan dan citra grayscale untuk menampilkan 15 citra pertama dari data MNIST tersebut

1. Lakukan proses ekstraksi fitur pada data MNIST. Pada proses ini, Anda diperbolehkan untuk
   * Langsung menggunakan nilai asli dari citra (nilai asli setiap pixel dari dataset) sebagai fitur.

Kode program :



Output :



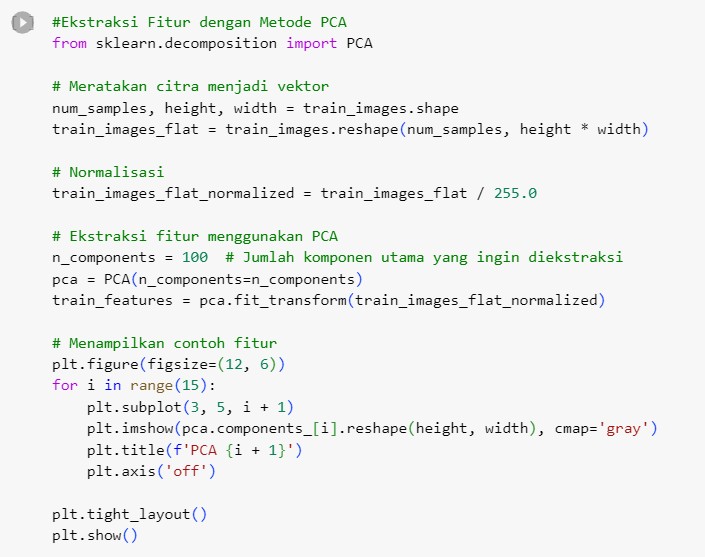
# Penjelasan:

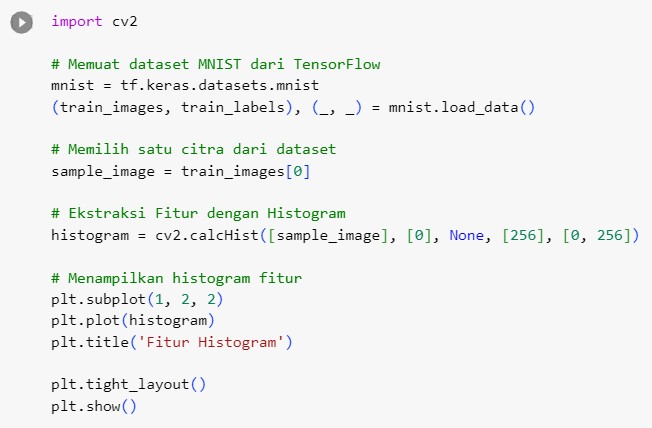
Kode program ini memiliki dua tujuan utama. Pertama, itu menampilkan dua gambar dalam satu jendela. Gambar pertama menampilkan citra digit tulisan tangan asli, sementara gambar kedua adalah grafik batang yang menggambarkan nilai piksel dari citra asli. Setiap bar pada grafik tersebut mewakili nilai piksel dalam citra tersebut.

Dengan kata lain, kode ini memvisualisasikan citra digit tulisan tangan dan kemudian menggambarkan representasi grafis dari nilai-nilai piksel dalam citra tersebut. Hal ini membantu pengguna untuk melihat bagaimana nilai piksel asli dari citra tersebut diwakili dalam bentuk grafik batang, sehingga memungkinkan pemahaman yang lebih baik tentang struktur dan karakteristik citra.

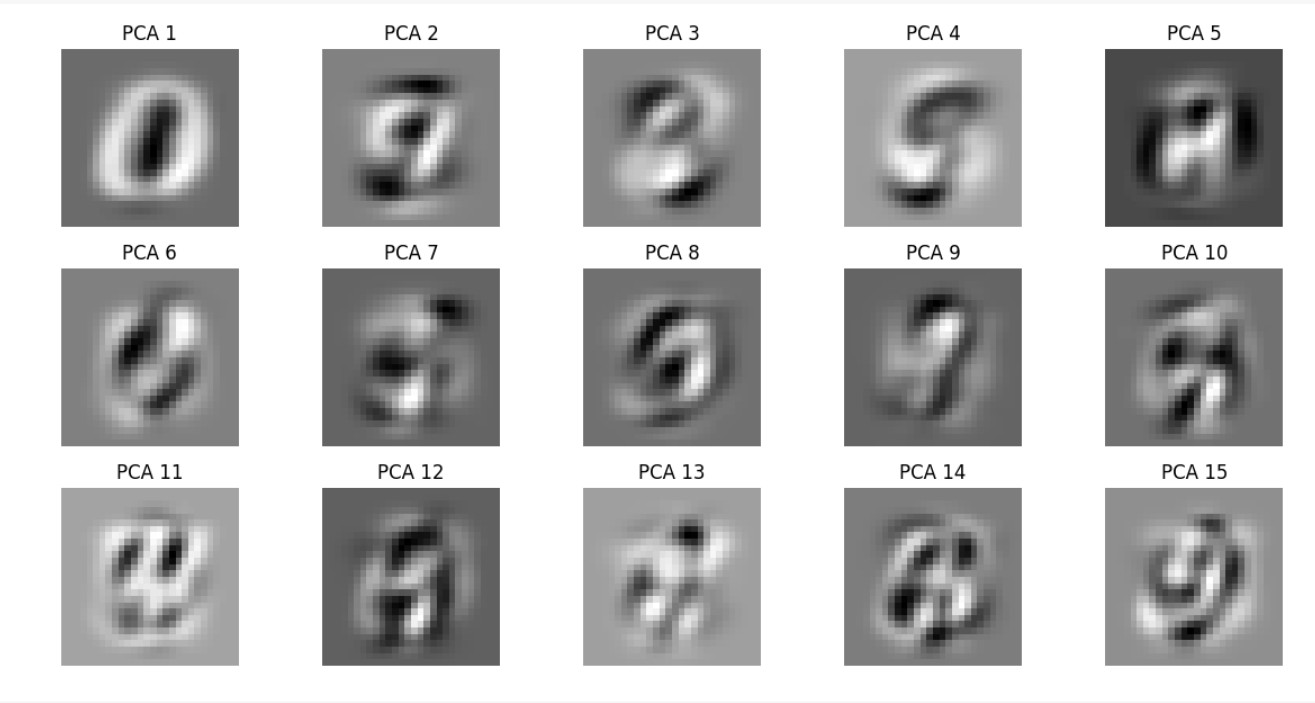
* + Melakukan ekstraksi fitur lain seperti histogram, PCA, atau yang lainnya. Anda diperbolehkan melakukan ekspolarasi pada proses ini.

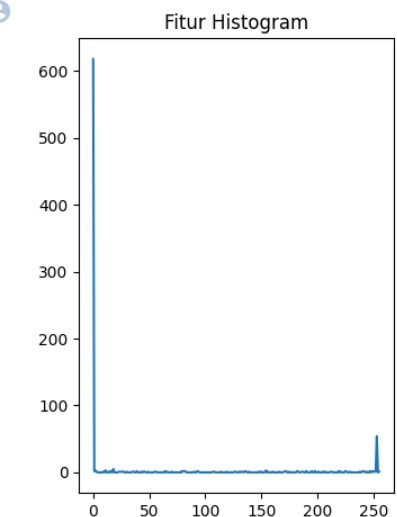
Kode program :





Output :





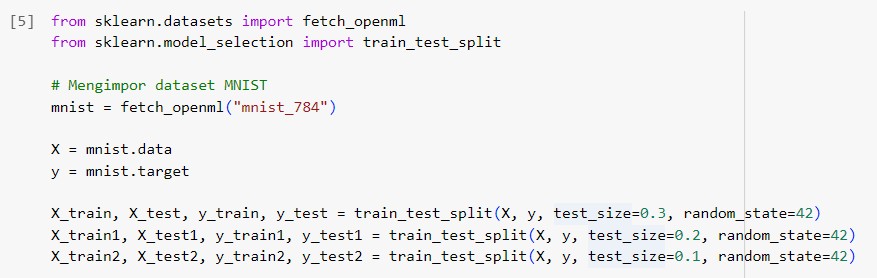
# Penjelasan:

* Kode program di atas akan menampilkan 15 gambar dalam satu jendela. Setiap gambar adalah representasi visual dari komponen utama yang diekstraksi dari data citra MNIST. Komponen utama adalah cara statistik untuk merepresentasikan variasi utama dalam data, sehingga gambar-gambar tersebut mencoba menggambarkan bagian penting dari variasi dalam

citra-citra MNIST.

* Kode program ini akan menampilkan satu plot yang merupakan histogram dari citra pertama dalam dataset MNIST. Histogram menggambarkan distribusi intensitas piksel dalam citra tersebut. Anda akan melihat seberapa sering intensitas piksel tertentu muncul dalam citra.

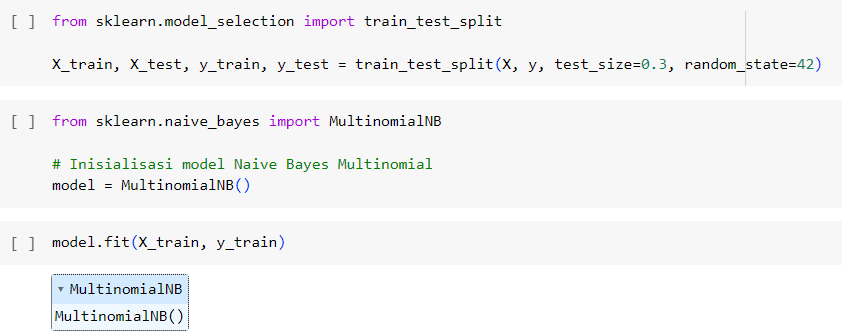
1. Buatlah data training dan data testing menggunakan rasio 70:30, 80:20, 90:10.



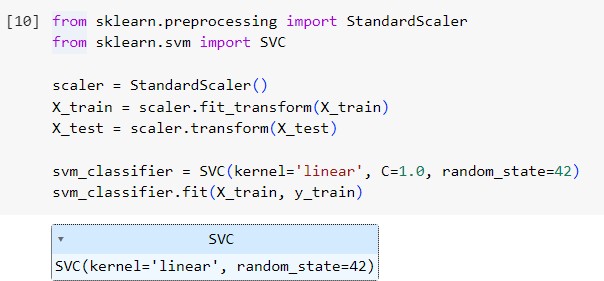
* + test\_size = 0.3 artinya rasio 70:30, test\_size = 0.2 artinya rasio 80:20, test\_size = 0.1

artinya rasio 90:10

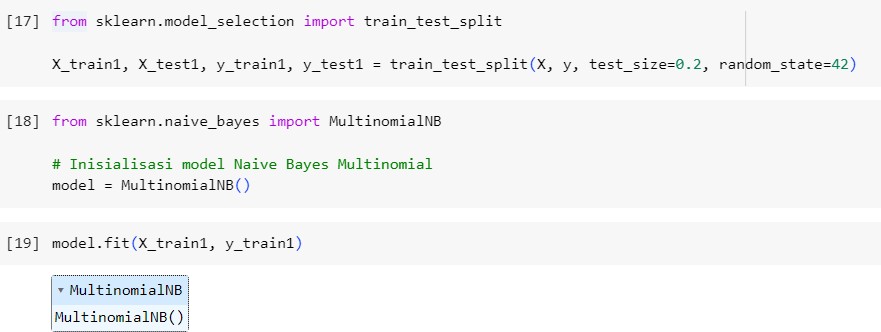
1. Lakukan proses klasifikasi dengan menggunakan algoritma Naive Bayes dan SVM.
   * Anda diperbolehkan untuk melakukan tuning parameter.
   * Anda diperbolehkan mengeksplorasi jenis kernel yang digunakan pada SVM.
2. Rasio 70:30
   * Naive Bayes



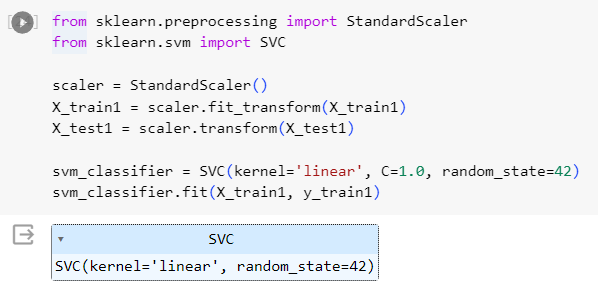
* + SVM



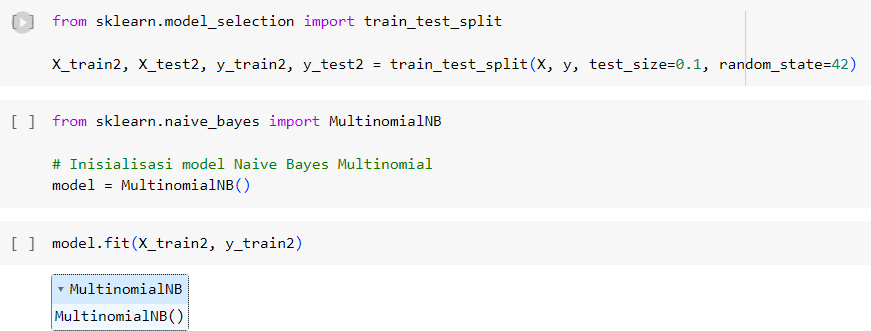
1. Rasio 80:20
   * Naive Bayes



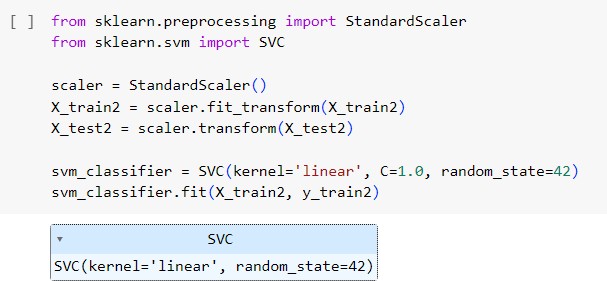
* + SVM



1. Rasio 90:10
   * Naive Bayes



* + SVM



# Penjelasan:

Kode program di atas digunakan untuk melakukan klasifikasi menggunakan dua algoritma, yaitu Naive Bayes dan Support Vector Machine (SVM), pada dataset MNIST yang telah diolah sebelumnya. Berikut adalah penjelasan singkat tentang kode tersebut:

Bagian Pertama (Naive Bayes):

* + Kode program dimulai dengan mengimpor library yang diperlukan, seperti library untuk Naive Bayes, SVM, dan pemrosesan data.
  + Dataset MNIST dibagi menjadi dua bagian: data pelatihan dan data pengujian dengan rasio yang telah ditentukan.
  + Model Naive Bayes diinisialisasi dan dilatih dengan data pelatihan.
  + Setelah pelatihan, model Naive Bayes digunakan untuk melakukan prediksi pada data pengujian.
  + Akurasi dari hasil prediksi diukur dan dievaluasi.

Bagian Kedua (SVM dengan Kernel Polinomial):

* + Pada bagian ini, dataset MNIST diunduh dan data serta label diekstrak.
  + Data kemudian dibagi menjadi data pelatihan dan data pengujian seperti pada bagian pertama.
  + Model SVM dengan kernel polinomial diinisialisasi dan dilatih dengan data pelatihan.
  + Setelah pelatihan, model SVM digunakan untuk memprediksi label data pengujian.
  + Akurasi hasil prediksi diukur dan dievaluasi.

Kesimpulannya, kode ini memungkinkan untuk melakukan tunning parameter pada kedua algoritma klasifikasi, Naive Bayes dan SVM, untuk mencari model yang optimal dalam setiap kasus. Akurasi dari hasil prediksi dari masing-masing model digunakan sebagai metrik untuk mengevaluasi kinerja klasifikasi pada dataset MNIST yang telah diproses sebelumnya. Dengan demikian, kode ini membantu dalam memahami dan membandingkan kinerja kedua algoritma tersebut dalam konteks dataset MNIST.

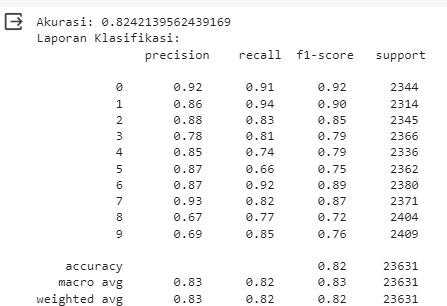
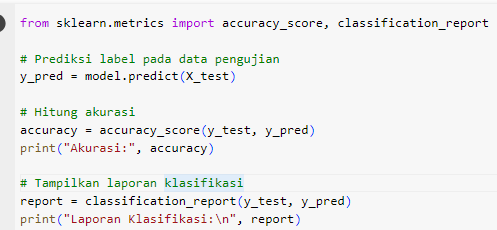
1. Evaluasi model yang Anda buat terhadap data training dan data testing. Gunakan metric

*accuracy* untuk mengetahui tingkat akurasi data training dan data testing.

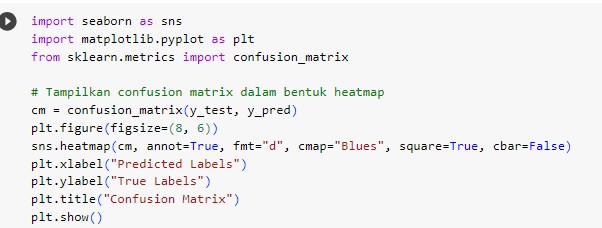
* + Gunakan fungsi *classification report* untuk mengetahui performa model secara holistik.
  + Gunakan confusion matrix untuk mengetahui tingkat ketepatan pelabelan.

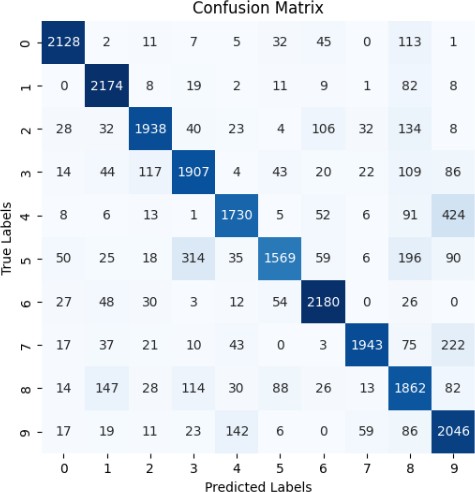
1. Rasio 70:30
   * Naive Bayes

→ Classification report and Accuracy



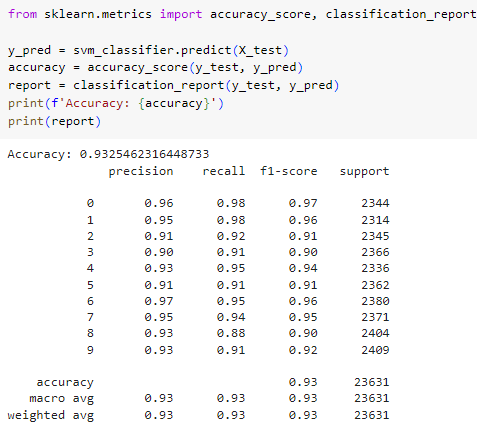
→ Confusion matrix





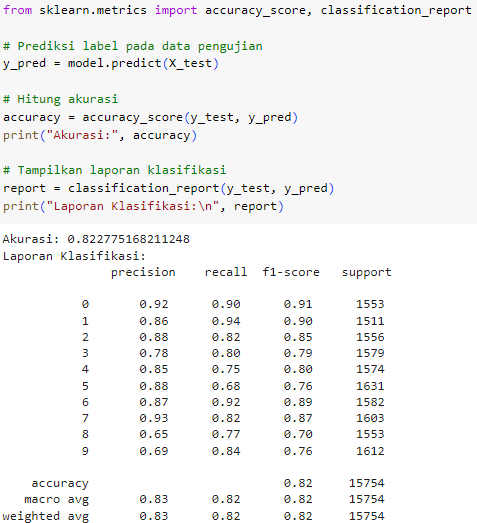
* + SVM

→ Classification report and Accuracy

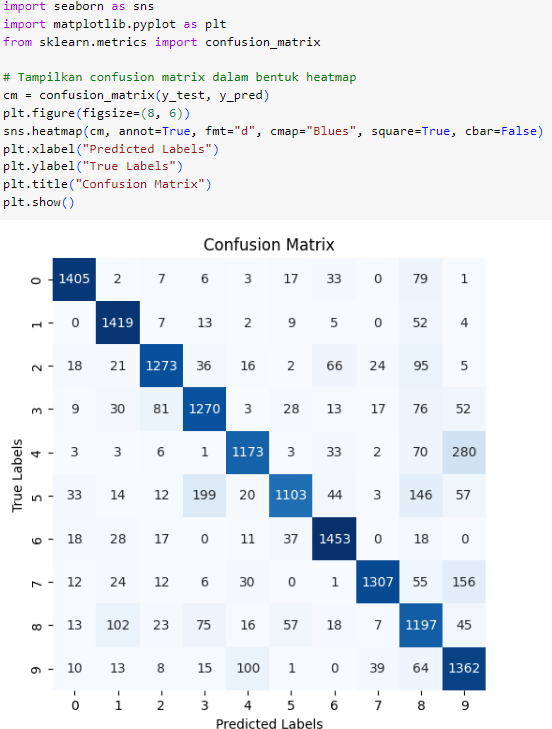


1. Rasio 80:20
   * Naive Bayes

→ Classification report and Accuracy

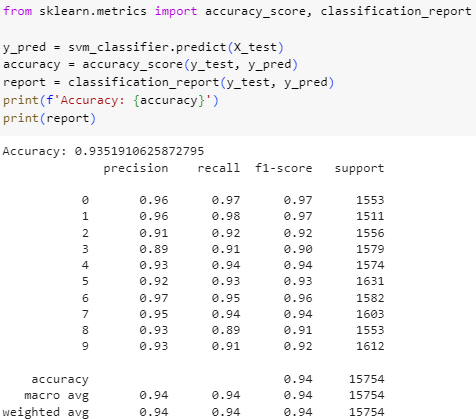


→ Confusion matrix



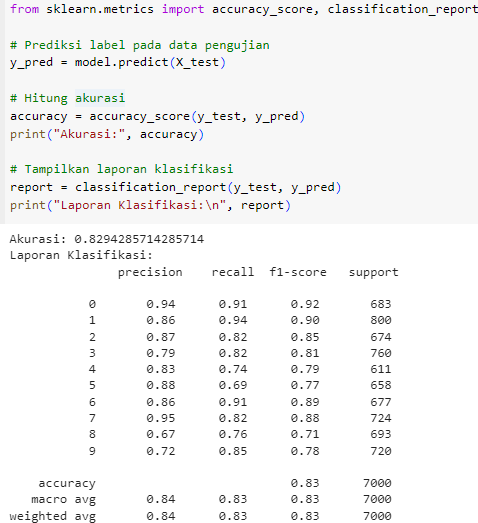
* + SVM

→ Classification report and Accuracy

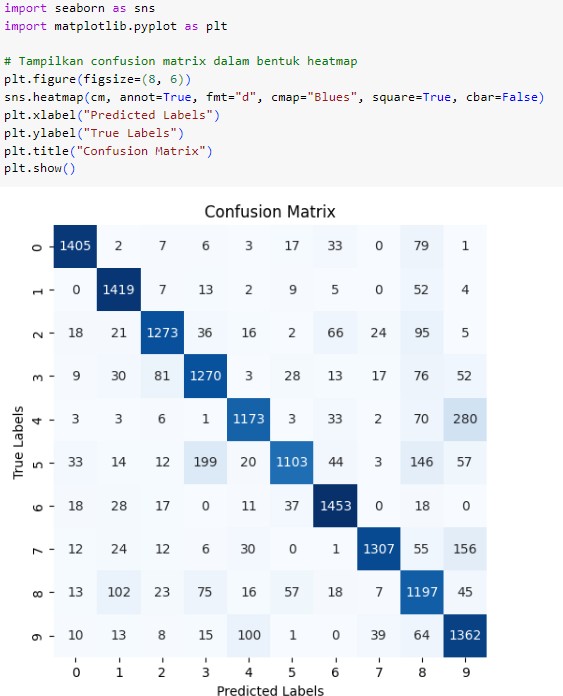


1. Rasio 90:10
   * Naive Bayes

→ Classification report and Accuracy

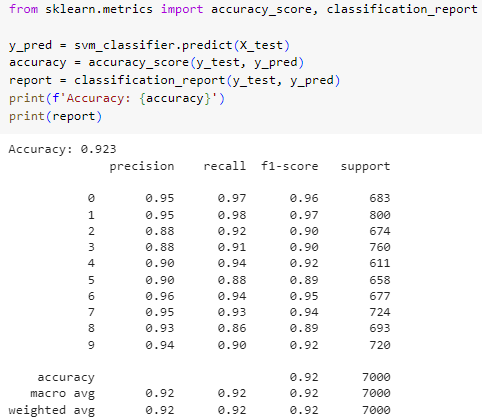


→ Confusion matrix



* + SVM

→ Classification report and Accuracy

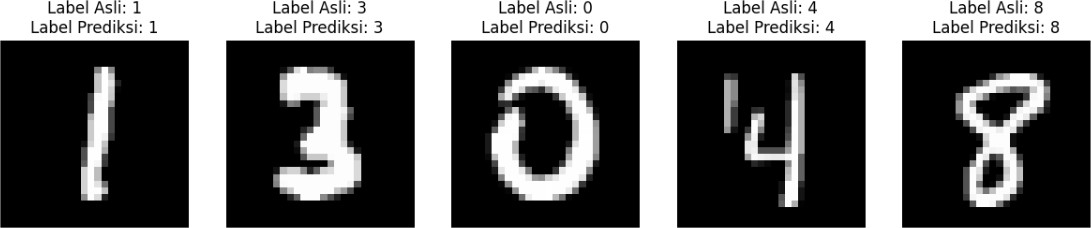
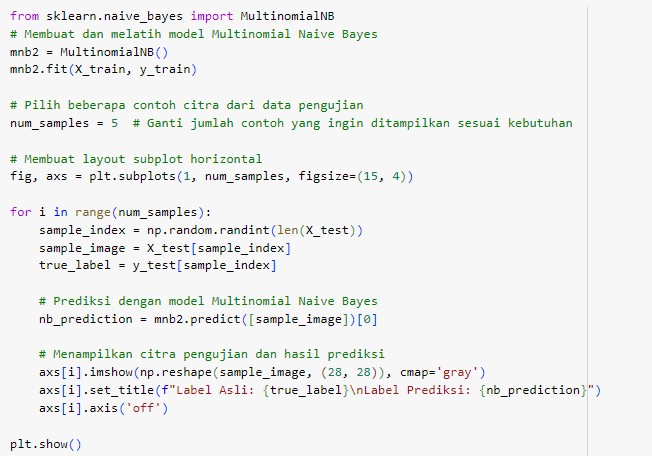


# Penjelasan:

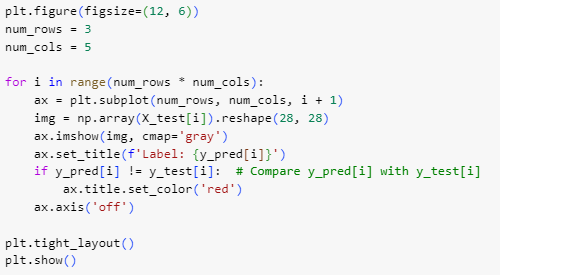
Dalam evaluasi model yang telah dibuat terhadap data pelatihan dan pengujian, metrik akurasi digunakan untuk mengukur tingkat ketepatan prediksi model pada kedua jenis data tersebut. Selain akurasi, laporan klasifikasi juga digunakan untuk memberikan gambaran holistik tentang kinerja model. Hasil laporan ini menunjukkan bahwa model memiliki presisi, recall, f1-score dan support yang tinggi untuk setiap kelas, menunjukkan bahwa model jarang membuat kesalahan dalam mengklasifikasikan data. Keseluruhan, hasil evaluasi menunjukkan bahwa model SVM yang telah dibuat sangat efektif dalam mengatasi permasalahan klasifikasi pada dataset yang digunakan

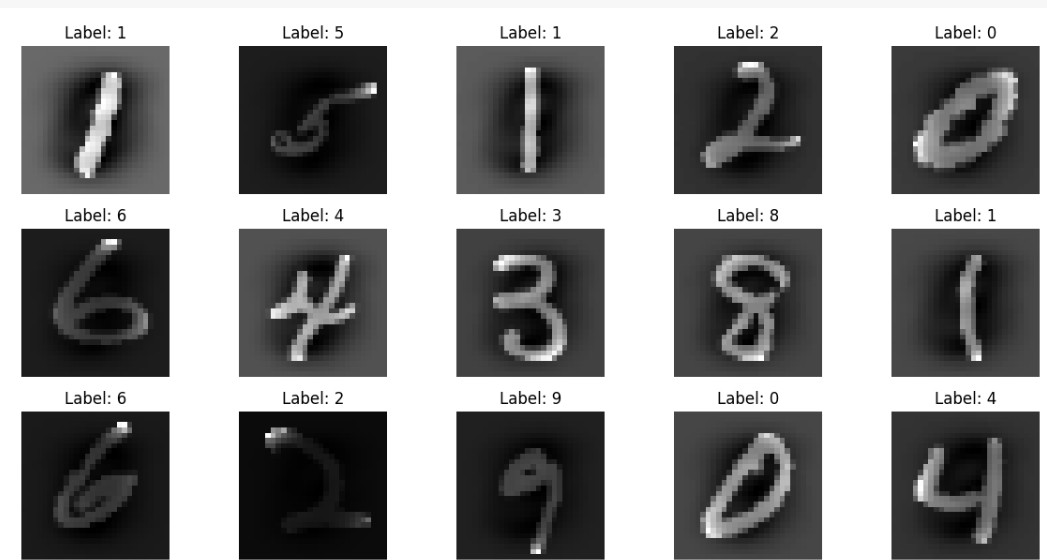
.

1. Tampilkan citra data testing beserta hasil pelabelannya. Anda dapat menggunakan Job Sheet 03 sebagai acuan.
2. Rasio 70:30
   * Naive Bayes

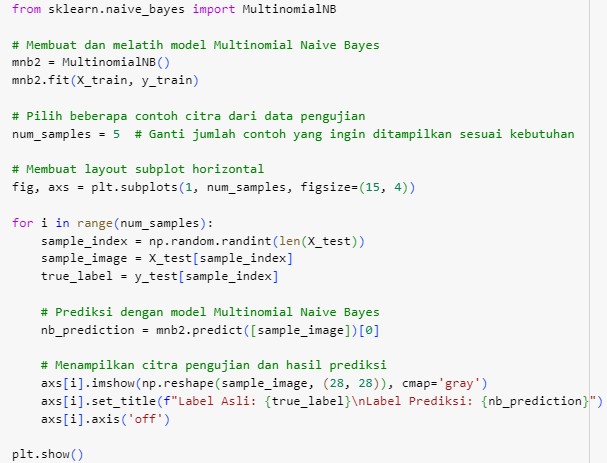


* + SVM

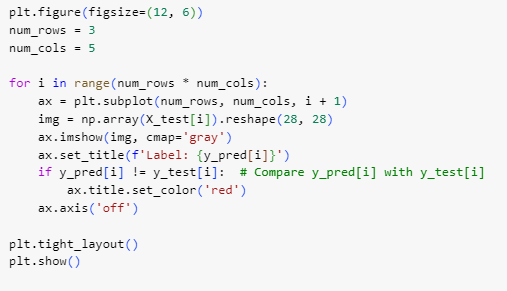


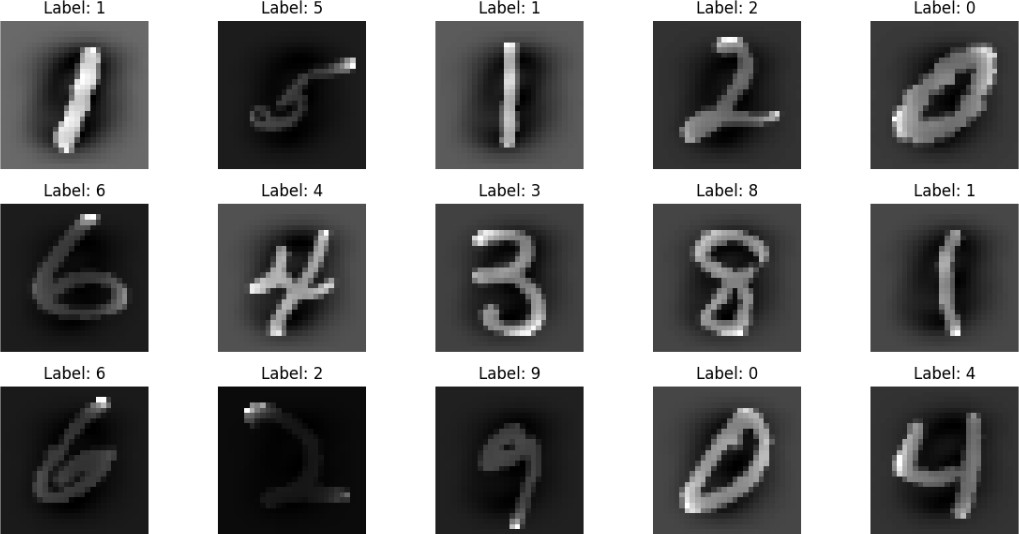


1. Rasio 80:20
   * Naive Bayes

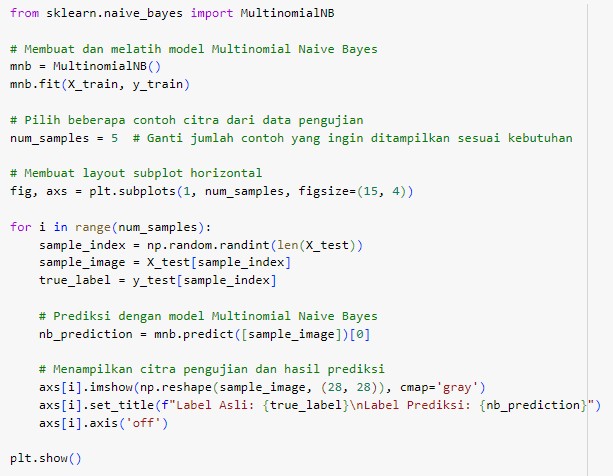


* + SVM

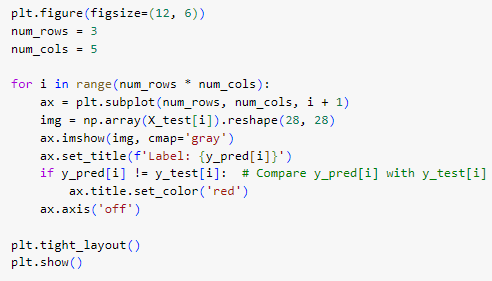


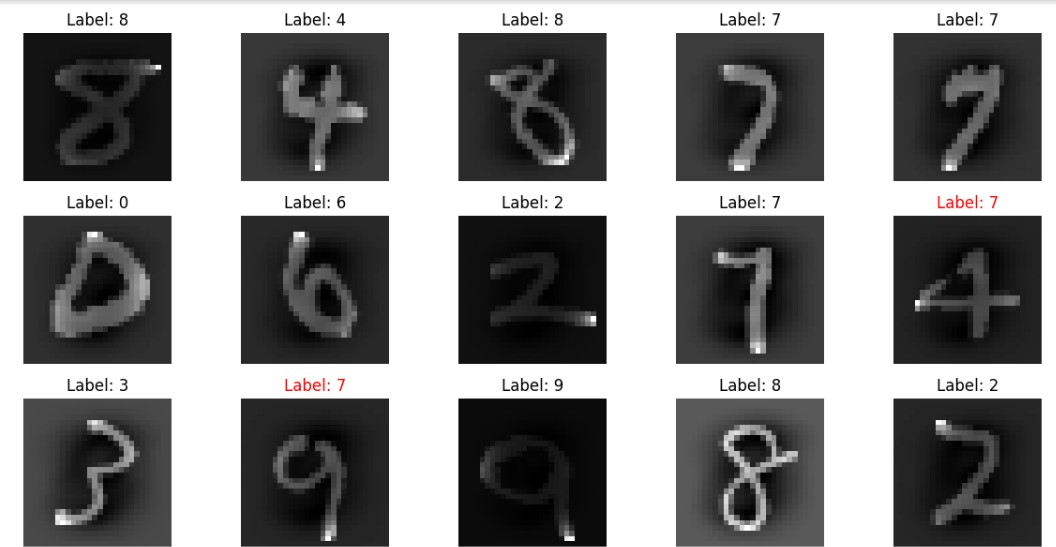


1. Rasio 90:10
   * Naive Bayes



* + SVM





1. Apa model terbaik yang Anda dapatkan? Bagaimana konfigurasinya? Berapa tingkat akurasi yang didapatkan? Jelaskan!

Dari perbandingan akurasi yang terdapat pada Soal no 6, klasifikasi data MNIST menggunakan model Naive Bayes dan SVM, kita dapat menyimpulkan hasil sebagai berikut:

* Model Terbaik:

Model SVM (Support Vector Machine) dengan kernel linear menghasilkan tingkat akurasi yang lebih tinggi dibandingkan dengan model Naive Bayes pada data MNIST.

* Konfigurasi Model Terbaik (SVM):

Model SVM menggunakan kernel linear.

* Tingkat Akurasi Model

Support Vector Machine (SVM)

- SVM1(70:30): Akurasi SVM = 0.9325462316448733

- SVM2(80:20): Akurasi SVM = 0.9351910625872795

- SVM3(90:10): Akurasi SVM = 0.923

Naive Bayes

- NaiveBayes(70:30): Akurasi data uji = 0.8242139562439169

- NaiveBayes2(80:20): Akurasi data uji = 0.822775168211248

- NaiveBayes3(90:10): Akurasi data uji = 0.8294285714285714

* Penjelasan:
  + Model SVM dengan kernel linear cenderung menjadi pilihan yang lebih baik dalam tugas klasifikasi data MNIST dibandingkan dengan Naive Bayes. Hal ini disebabkan oleh kemampuan SVM untuk menangani masalah klasifikasi dengan baik, terutama pada data yang memiliki dimensi tinggi seperti citra MNIST.
  + Dari hasil di atas, terlihat bahwa SVM mengungguli Naive Bayes dalam semua tiga konfigurasi yang berbeda. Oleh karena itu, SVM adalah model terbaik dalam kasus ini. dalam kasus ini, SVM adalah model terbaik dengan tingkat akurasi tertinggi (0.9351), diikuti oleh NaiveBayes2 dengan tingkat akurasi yang lebih rendah (0.8227).