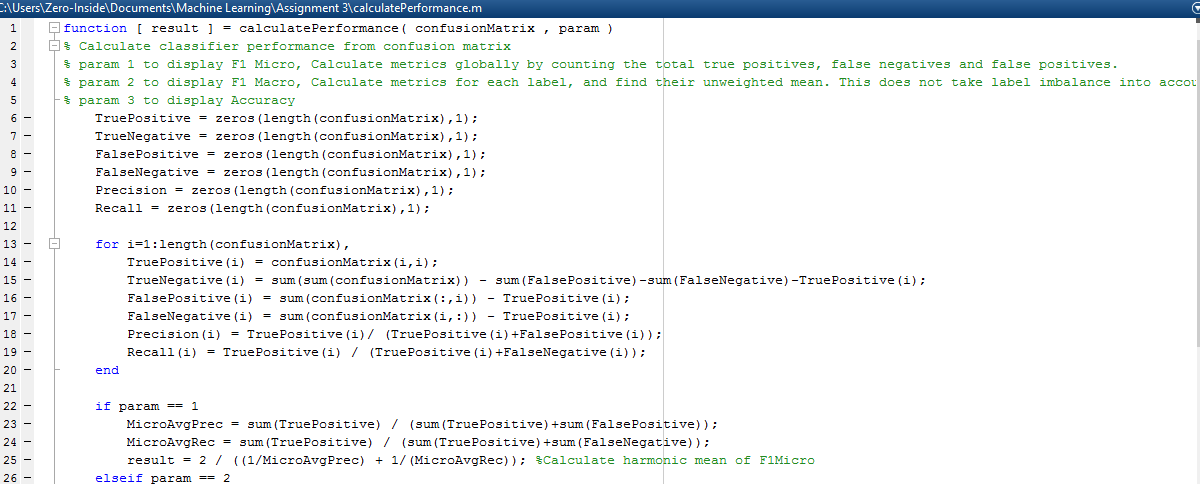
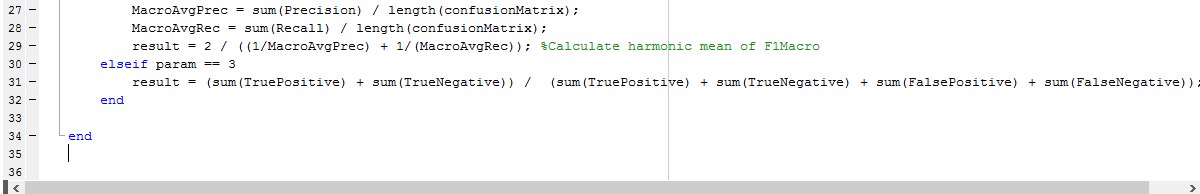
1. Calculate Performance

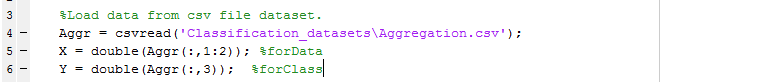


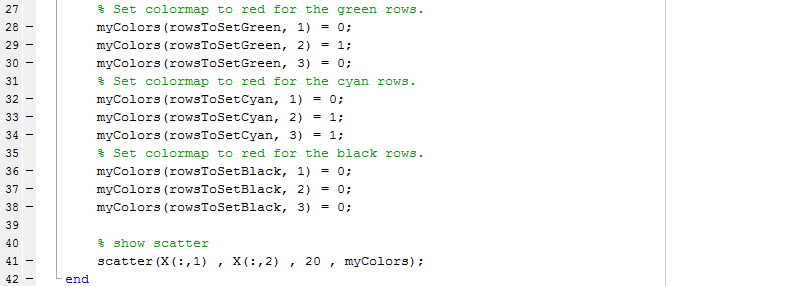
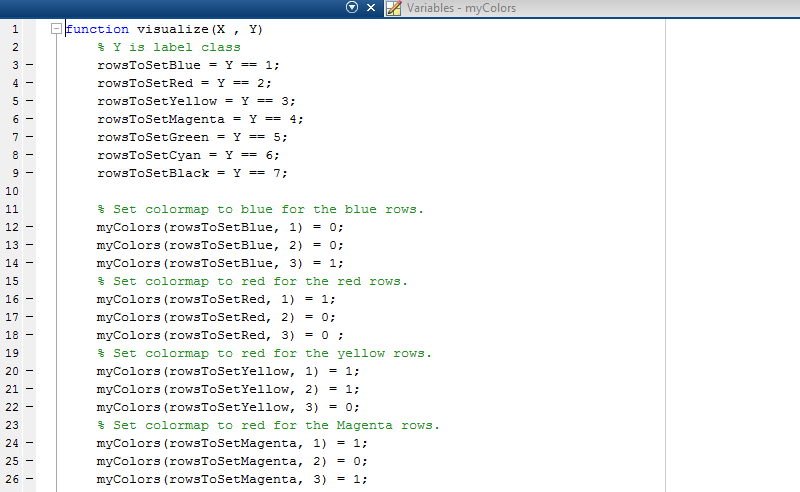


Analisis Program :

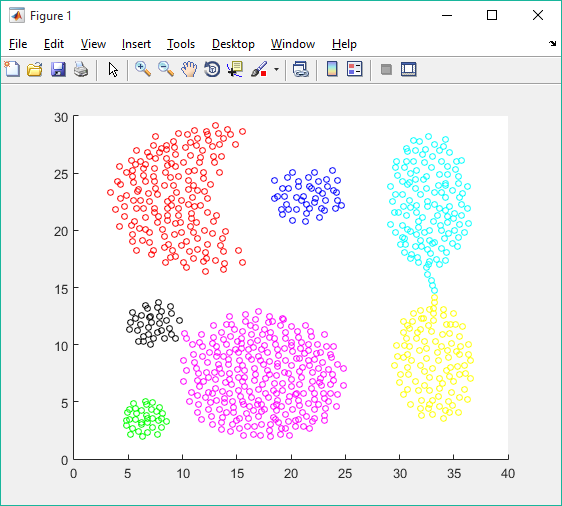
Fungsi calculatePerformance untuk menghitung performansi dari classifier dengan inputan berupa confusion matrix dan param yaitu jenis kalkulasi yang ingin ditampilkan, jika param adalah 1 maka menghitung F1 Micro, jika param adalah 2 maka menghitung F1 Macro dan jika param adalah 3 maka menghitung accuracy.

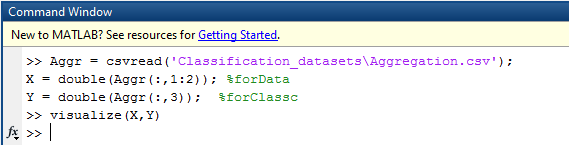
1. Naïve Bayes
2. Load & Visualize Data





Output :

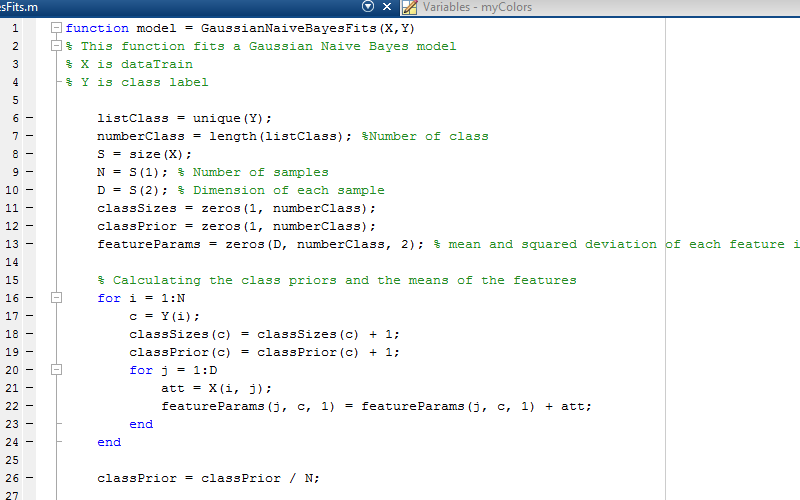


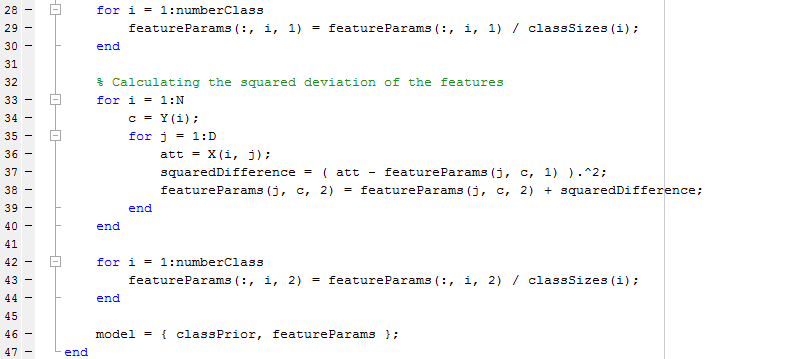


Analisis Program :

Dengan menggunakan scatter, memvisualisasikan dataset lebih mudah. Terlebih dahulu meload dataset dan membagi data train dan data class label. Kemudian memanggil funsi visualize dengan parameter data dan label. Pada fungsi visualize(x,y) masing-masing data label class disesuaikan dengan data warna RGB. Kemudian memanggil scatter(param1,param2,param3,param4) dengan param1 dan param2 adalah kolom pertama dari data train dan kolom kedua dari data train, kemudian param3 adalah size dari ukuran plot, dan param4 adalah data color.

1. Apply Naïve Bayes
2. Function for Learning

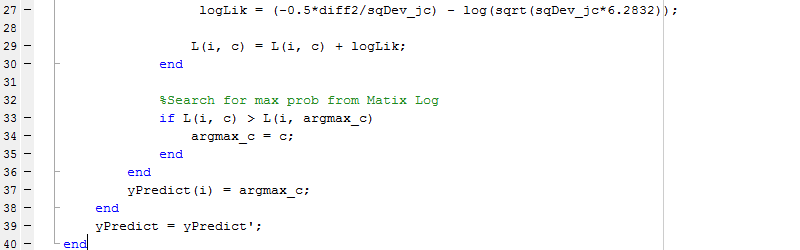
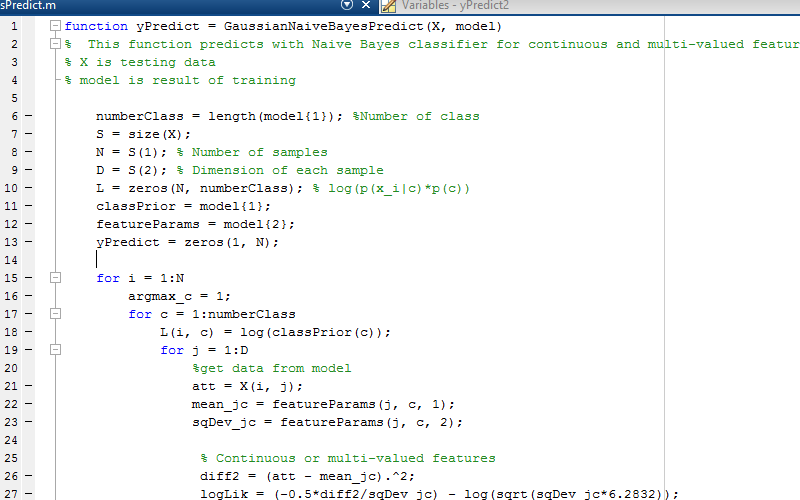




Analisis Program :

Fungsi ini membuat sebuah variable objek dari hasil training. Objek tersebut terdiri dari 2 kolom, yang pertama array class prior yang didapat dari menghitung semua class yang sama pada label dan membaginya dengan total dari y yaitu 788, maka akan didapat prior masing-masing class yaitu 0.0571065989847716 , 0.215736040609137 , 0.129441624365482 , 0.346446700507614, 0.0431472081218274 , 0.164974619289340, 0.0431472081218274. yang kedua adalah likelihood yang terdiri dari array mean dan standart deviasi pada setiap kelas.

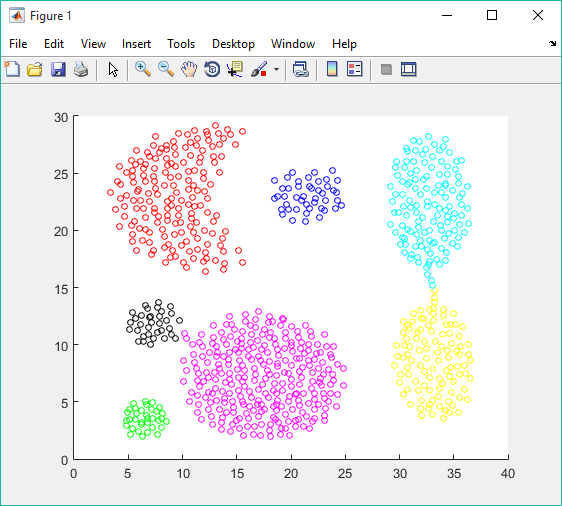
1. Function for Predict

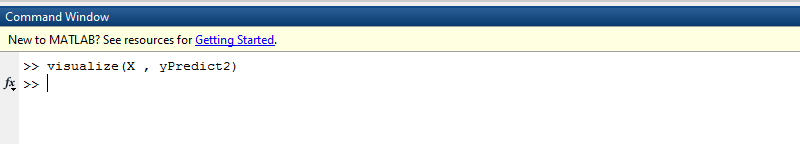


Analisis Program :

Fungsi ini mengeluarkan sebuah vector dari hasil prediksi berisi label class. Label class didapat dari perhitungan data testing dengan data model yang sudah dihasilkan melalui fungsi GaussianNaiveBayesFits. Proses perhitungan posterior yaitu dengan menambahkan hasil log pada classPrior dengan log (-0.5\*diff2/sqDev\_jc) - log(sqrt(sqDev\_jc\*6.2832)). Dimana sqDev\_jc adalah standart deviasi dari model, dan diff2 diperoleh dengan mengurangkan nilai kolom dengan mean kolom yang ada pada model. Kemudian untuk menentukan class dilakukan perulangan pada setiap data, kemudian membandingkan nilai log setiap kelas mana yang lebih besar maka itu kelasnya.

1. Plot result of Predict





Analisis Program :

Dengan memanggil visualize(x,yPredict2), dimana x adalah train data dan yPredict adalah class yang dihasilkan melalui testing maka ditampilkan scatter yang memplot posisi dari data train.

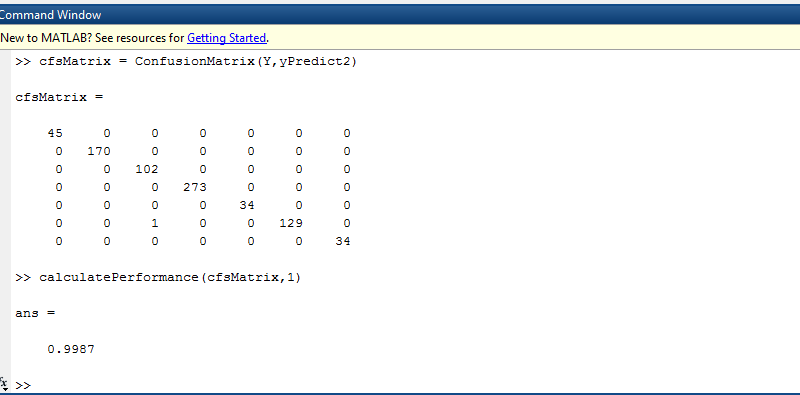
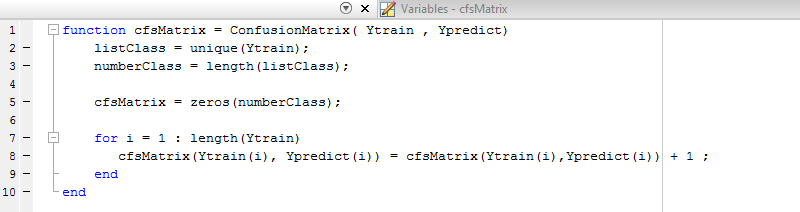
1. Compare Visualize

|  |  |
| --- | --- |
| Scatter Data Asli | Scatter Data Prediksi |
|  |  |

Analisis :

Terlihat ada satu warna yang berubah pada class cyan (6) terhadap class yellow(3). Tetapi secara keseluruhan naïve bayes dapat mengkelompokkan data tersebut sesuai kelasnya dalam visual ini adalah warna.

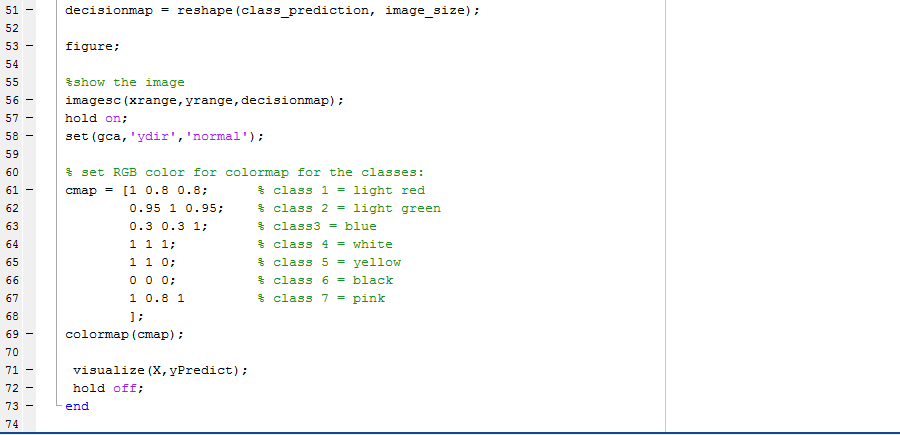
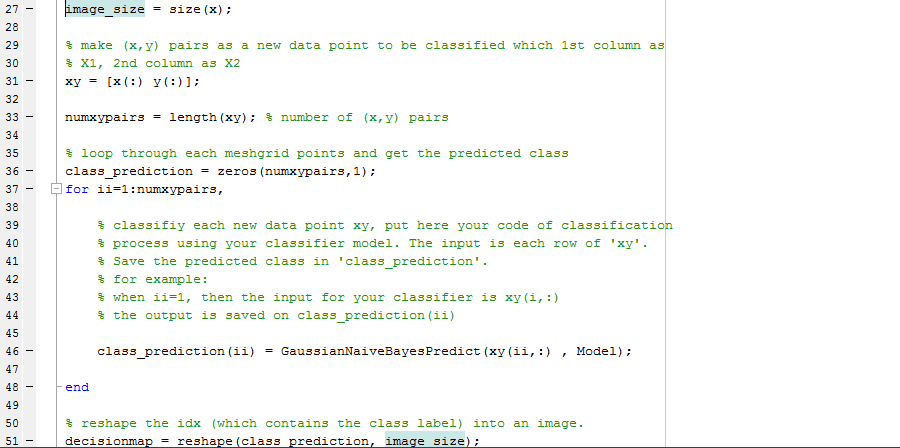
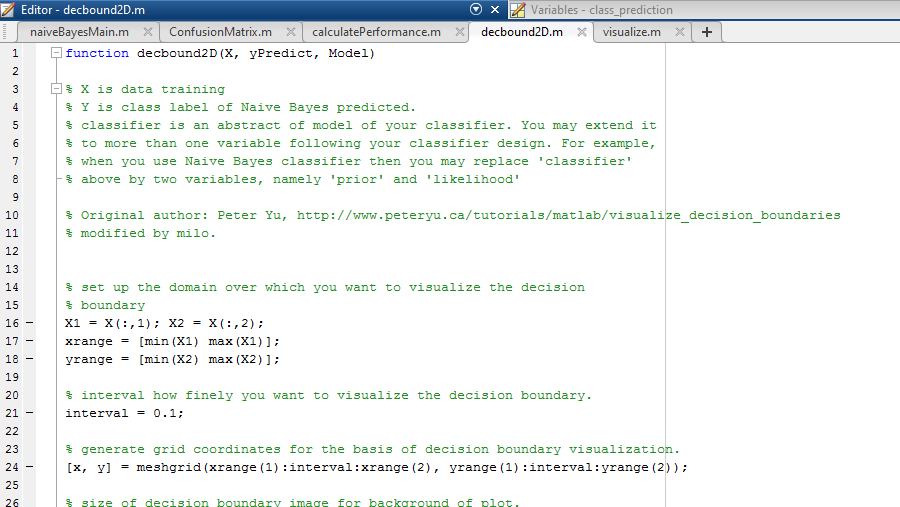
1. Calculate Performance



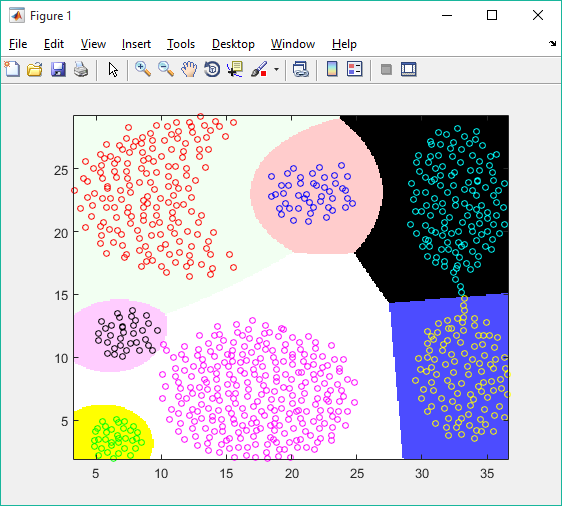
Analisis Program :

Untuk menghitung performance terlebih dahulu membuat confusion matrix berdasarkan label data train dan label data testing. Kemudian untuk menghitung performasi memanggil fungsi calculatePerformance(param1, param2). Dengan param1 adalah confusion matrix dan param2 adalah mode atau jenis performasi, 1 untuk F1 Micro 2 untuk F2 Macro dan 3 untuk Accuracy. Berdasarkan hasil perhitungan didapat hasil klasifikasi naïve bayes adalah 0.9987. atau dalam persen adalah 99.87%.

1. Plot Decision Boundary



Output :



Analisis Program :

Fungsi decbound2d(param1,param2,param3) akan menampilkan decision boundaries dengan param1 adalah data testing , param2 adalah class label hasil dari naïve bayes, dan param3 adalah model yang akan digunakan untuk memberi label kelas pada new data point yang dipetakan pada imagesec.