

Laporan Tugas Program 4 Mesin Pembelajaran

Ahmad Arsyel Abdul Hakim / 1301164193

IF-40-06

A. Analisis Masalah

Permasalahan dalam tugas program 4 pembelajaran mesin ini adalah menentukan dua label/klasifikasi (bernilai 1 dan 2) yang diberikan dari 298 objek data latih terhadap 75 objek data tes. Objek yang diberikan memiliki dua atribut (x dan y) dengan satu *class*. Data tersebut disajikan dalam file *.csv* dan akan dipilih algoritma *Naïve Bayes* dan algoritma tertentu, sehingga label/*class* pada data tes dapat ditentukan.

B. Strategi Penyelesaian

Berdasarkan analisis di atas, algoritma yang dapat digunakan untuk menyelesaikan permasalahan tersebut adalah dengan *Ensemble Learning: Bagging* berbasis *Naïve Bayes*.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Gambar 1. Rumus Gaussian.

Untuk menerapkan *bagging*, diperlukan data latih (*trainset*) yang baru dengan melakukan bootstrapping sebanyak model yang diinginkan. Banyaknya model dapat mempengaruhi akurasi dari hasil akhir. Setelah itu, lakukan implementasi model terhadap *datatest* menggunakan *Naïve Bayes* dengan perhitungan kontinu. Dilakukan penerapan rumus *Gaussian* seperti pada Gambar 1 dan algoritmanya pada Gambar 4.

Dalam perhitungan *Gaussian*, terdapat nilai standar deviasi yang harus terdefinisi, maka dari itu algoritma standar deviasi diimplementasikan seperti pada Gambar 3. Hasil dari model yang baru, di terapkan pada *testset*. Model yang dipilih sebanyak 21, karena akurasi yang sudah dicoba dari *datatrain* ketika jadi *datatest* sebesar ~94.63 %.

C. Penjelasan Fungsi

Ada beberapa fungsi krusial yang dipakai dalam kode program ini. Yaitu pada fungsi *naïve_bayes*, isi parameter meminta masukan *datatrain* dan *datatest*, sehingga output yang diberikan berupa pemberian label pada *datatest*.

Algoritma bootstrap juga diimplementasi seperti pada Gambar 5. Dengan melakukan *generate* data sebanyak *nmodel*.

```
local function naive_bayes(tab_train, tab_test)
```

Gambar 2. Fungsi Naïve Bayes.

```
local function std(tab, col_name, avg, n, label)
  local result = 0
  local dt
  for _, v in pairs(tab) do
    if (v.label == label) then
      dt = v[col_name] - avg
      result = result + dt^2
    end
  end
  return math.sqrt(result / (n-1))
end
```

Gambar 3. Algoritma menghitung standar deviasi.

```

local function p_count(stdev, curr_val, avg)
  return 1 * math.exp(-(curr_val - avg)^2 / (2 * (stdev)^2)) /
    (stdev * math.sqrt(2 * math.pi))
end

```

Gambar 4. Menentukan nilai probabilitas.

```

local function create_bootstraps(tab, n)
  local bootstraps = {}
  for i=1, n do
    bootstraps[i] = {}
    for j=1, #tab do
      bootstraps[i][j] = {}
      for k, v in pairs(tab[math.random(1, #tab)]) do
        bootstraps[i][j][k] = v
      end
    end
    table.sort(bootstraps[i], function(f_val, s_val) return f_val.id < s_val.id end)
    --print_table(bootstraps[i])
  end
  return bootstraps
end

```

Gambar 5. Membuat bootstrap dari data latih.

```

--> BAGGIN
local nmodel = 21 -- optimal accuracy in ~94.630872483221 %

      ACCURACY:      93.624161073826 %

```

Gambar 6. Banyaknya model yang dipilih dan hasil akurasi.