

Nama : Khulafaur Rasyidin

Kelas : IFX-42-01

NIM : 1301188571

Kita diberikan sebuah Trainset berupa himpunan data berisi 160 objek data yang memiliki 7 atribut input (**age, workclass, education, marital-status, occupation, relationship, hours-per-week**) dan 1 output (label kelas **income**) yang memiliki 2 kelas/label (>50K, dan <=50K).

Lalu kemudian, kita diminta untuk membangun system klasifikasi menggunakan metode Naïve Bayes untuk menentukan kelas/label data testing dalam Testset

Menurut Olson Delen (2008) menjelaskan Naïve Bayes untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informasi objek. Algoritma ini mengasumsikan bahwa atribut objek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari "master" tabel keputusan.

Tahapan proses Naive Bayes

1. Menghitung jumlah kelas / label
2. Menghitung jumlah kasus per-kelas
3. Kalikan semua variabel kelas
4. Bandingkan hasil per kelas

Keuntungan penggunaan adalah bahwa metoda ini hanya membutuhkan jumlah data pelatihan (training data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel independen, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

Mari kita lihat implementasinya dalam bentuk program yang dibuat menggunakan Bahasa Python

1. Kita baca file dataset dan buat fungsinya untuk diproses

```
2
3 # Baca TrainSet dan TestSet
4 Train = pd.read_csv('TrainsetTugas1ML.csv')
5 Test = pd.read_csv('TestsetTugas1ML.csv')
6
7
8 # Kita buat fungsinya
9 def count(lab1, value1, lab2,
10          value2):
11     i = 0
12     for j in range(len(Train)):
13         if (Train[lab1][j] == value1 & (Train[lab2][j] == value2):
14             i = i + 1
15     return i
```

2. Kita cari peluangnya

```
17 #Kita hitung peluangnya menggunakan naive bayes
18 def naivebayes(lab1, value1, lab2, value2, income_type):
19     result = count(lab1, value1, lab2, value2) / income_type
20     return result
```

Nama : Khulafaur Rasyidin

Kelas : IFX-42-01

NIM : 1301188571

3. Tentukan peluang incomenya

```
22 # Langkah Pertama Kita Tentukan Peluang Incomenya -> >50K dan <=50K
23 big_income = 0 # ngitung banyak income >50K
24 small_income = 0 # ngitung banyak income <=50K
25 for a in range(len(Train)):
26     if (Train['income'][a] == '>50K'):
27         big_income = big_income + 1
28     else:
29         small_income = small_income + 1
30 Pelbig_income = big_income / len(Train)
31 Pelsmall_income = small_income / len(Train)
```

4. Cari peluang dari masing-masing atribut

```
33 # Langkah Kedua Kita Tentukan Peluang Dari Masing-Masing Atribut
34 income = []
35 for i in range(len(Test)):
36     PelageBig = naivebayes('age', Test['age'][i], 'income', '>50K', big_income)
37     PelageSmall = naivebayes('age', Test['age'][i], 'income', '<=50K', small_income)
38
39     PelworkclassBig = naivebayes('workclass', Test['workclass'][i], 'income', '>50K', big_income)
40     PelworkclassSmall = naivebayes('workclass', Test['workclass'][i], 'income', '<=50K', small_income)
41
42     PeleducationBig = naivebayes('education', Test['education'][i], 'income', '>50K', big_income)
43     PeleducationSmall = naivebayes('education', Test['education'][i], 'income', '<=50K', small_income)
44
45     PelmaritalstatusBig = naivebayes('marital-status', Test['marital-status'][i], 'income', '>50K', big_income)
46     PelmaritalstatusSmall = naivebayes('marital-status', Test['marital-status'][i], 'income', '<=50K', small_income)
47
48     PeloccupationBig = naivebayes('occupation', Test['occupation'][i], 'income', '>50K', big_income)
49     PeloccupationSmall = naivebayes('occupation', Test['occupation'][i], 'income', '<=50K', small_income)
50
51     PelrelationshipBig = naivebayes('relationship', Test['relationship'][i], 'income', '>50K', big_income)
52     PelrelationshipSmall = naivebayes('relationship', Test['relationship'][i], 'income', '<=50K', small_income)
53
54     PelhoursperweekBig = naivebayes('hours-per-week', Test['hours-per-week'][i], 'income', '>50K', big_income)
55     PelhoursperweekSmall = naivebayes('hours-per-week', Test['hours-per-week'][i], 'income', '<=50K', small_income)
56
57     Pup = PelageBig * PelworkclassBig * PeleducationBig * PelmaritalstatusBig * PeloccupationBig * PelrelationshipBig * PelhoursperweekBig
58     Pdown = PelageSmall * PelworkclassSmall * PeleducationSmall * PelmaritalstatusSmall * PeloccupationSmall * PelrelationshipSmall * PelhoursperweekSmall
59
```

5. Kita masukkan dan cetak hasilnya

```
66 # Terakhir Kita Masukkan Hasil Prediksi Kedalam File Datatest
67 for k in range(len(Test)):
68     age = []
69     workclass = []
70     education = []
71     marital = []
72     occupation = []
73     relationship = []
74     hours = []
75     age.append(Test['age'][k])
76     workclass.append(Test['workclass'][k])
77     education.append(Test['education'][k])
78     marital.append(Test['marital-status'][k])
79     occupation.append(Test['occupation'][k])
80     relationship.append(Test['relationship'][k])
81     hours.append(Test['hours-per-week'][k])
82
83 #Tampil dan Cetak Hasilnya
84 print(income)
85 output = pd.DataFrame(
86     {'age': age, 'workclass': workclass, 'education': education, 'marital-status': marital, 'occupation': occupation,
87      'relationship': relationship, 'hours-per-week': hours, 'income': income}, index=Test['id'])
88 output.to_csv("TebakanTugas1ML.csv") # Mengeuarkan output TebakanTugas1ML.csv
89
90
```

Hasilnya akan muncul dan membuat file baru TebakanTugas1ML.csv