Nama : Mela Mai Anggraini
NIM : 1301160307
Kelas : IF-40-09

Tugas 1.1 : Menentukan label income pada dataTest berdasarkan dataTrain yang ada menggunakan metode Naive Bayes.

Diketahui : pada dataTrain berisi 160 objek dan dataTest berisi 40 objek. Pada dataTrain memiliki 7 atribut input(age,workclass,education,marital-status,occupation,relationship,hours-perweek) dan 1 output label income yang memiliki 2 kelas yaitu '>50K' dan '<=50K'.

Step Solving:

1. Mencari banyaknya data pada dataTrain dan dataTest. Sehingga jika ada data baru maka program bisa handling data tersebut.
2. Dari informasi diatas maka,didapatkan rumus untuk menghitung peluang untuk setiap data pada dataTest sebagai berikut :

    - P('>50K'| X')= P(age|'>50K')* P(workclass|'>50K')* P(education|'>50K')* P(marital-status|'>50K')* P(occupation|'>50K')* P(relationship|'>50K')* P(hours=per-week|'>50K')*P(>50K)

    - P('>50K'| X')= P(age|' <=50K')* P(workclass|' <=50K')* P(education|' <=50K')* P(marital-status|' <=50K')* P(occupation|' <=50K')* P(relationship|' <=50K')* P(hours=per-week|'<=50K')*P('<=50K')

    Keterangan:
    - Misalkan: P(x|y)
        - x = data atribut dalam dataTest
        - y = data kelas income dalam dataTrain
            - Rumus : jumlah (atribut = x dan kelas income = y) / jumlah y
3. Jika salah satu hasil P('>50K'| X') atau P('>50K'| X') terdapat nilai 0 maka, gunakan metode Laplace Smoothing, dengan cara:
    - Setiap pembilang ditambah 1(+1)
    - Dan penyebut ditambah dengan jumlah kategori(unique value) yang ada dalam atribut.
        - Rumus: jumlah (atribut = x dan kelas income = y)+1 / jumlah y + jumlah kategori
4. Lakukan perbandingan antara hasil P('>50K'| X') dan P('>50K'| X').
    - Jika hasil P('>50K'| X') > P('>50K'| X') maka, data test tersebut masuk kedalam kelas income '>50K'.
    - Jika hasil P('>50K'| X') < P('>50K'| X') maka, data test tersebut masuk kedalam kelas income '<=50K'.
5. Lakukan step 2-4 pada **setiap** dataTest.
6. Setelah selesai, data tersebut di import ke dalam file 'TebakanTugas1ML.csv'

Fungsi yang digunakan:

- Cari_semua(a) : fungsi untuk menghitung jumlah a pada data train.

```
def cari_semua(a): #untuk menghitu
    hasil=0
    i=0
    while i<=160:
        if train[8][i] == a:
            hasil=hasil+1
        i=i+1
    return hasil
```

- Cari(a,b,c) : fungsi untuk menghitung jumlah data a yang memiliki income b didalam data train, dan c merupakan no kolom.

```
def cari(a,b,c): #untuk mencari data yang sesuai
    hasil=0
    i=0
    while i<=160:
        if train[c][i]== a and train[8][i]==b :
            hasil= hasil+1
        i=i+1
    return hasil
```

- Check_atribut(a): fungsi untuk mengembalikan nilai jumlah dari unique value(kategori) didalam suatu kolom/atribut.

```
def check_atribut(a): #fungsi untuk mengecheck didalam kolom ada berapa unique atribut
    list_atribut = set([atribut for atribut in train[1:][a]])
    return len(list_atribut)
```

Main Program:

```
while item<=len_test:
    x = cari(test[1][item],a,1)/income1*cari(test[2][item],a,2)/income1*cari(test[3][item],a,3)/income1*cari(test[4][item],a,4)/income1*cari(test[5][item],a,5)/income1*cari(test[6][i
    y = cari(test[1][item],b,1)/income2*cari(test[2][item],b,2)/income2*cari(test[3][item],b,3)/income2*cari(test[4][item],b,4)/income2*cari(test[5][item],b,5)/income2*cari(test[6][i
    if x==0 or y==0:
        #pakai laplace smoothing
        x= (cari(test[1][item],a,1)+1)/(income1+check_atribut(1))*(cari(test[2][item],a,2)+1)/(income1+check_atribut(2))*(cari(test[3][item],a,3)+1)/(income1+check_atribut(3))*(cari
        y= (cari(test[1][item],b,1)+1)/(income2+check_atribut(1))*(cari(test[2][item],b,2)+1)/(income2+check_atribut(2))*(cari(test[3][item],b,3)+1)/(income2+check_atribut(3))*(cari
    if x > y:
        result.append(a)
    else:
        result.append(b)
    item=item+1
df= pd.DataFrame(result)
df.to_csv('TebakanTugas1ML.csv',sep='\t',index=False,header=None)
```

*note: hanya sebagian yang bisa di screenshot, untuk lebih jelasnya silahkan liat source code.