Laporan Tugas 1.1 Naive Bayes

Oleh Kinegar Hadinawa

Analisis

Terdapat sebuah trainset berisi 160 data dengan memiliki 7 atribut input(age, workclass, education, marital-status, occupation, relationship, hours-per-week) dengan output income memiliki dua label (>50K dan <=50K). Tujuan dibuatnya program agar dapat menentukan class output jika diberi inputan dengan menggunakan metode Naive Bayes pada data baru. Digunakan probabilitas untuk menentukan output class sesuai 7 data data input.

Digunakan Naive Bayes Classification

$$P(X_1, X_2, \dots, X_n|C) = P(X_1|C)P(X_2|C) * \dots * P(X_n|C)$$

Dengan ini ditentukan class output yang memiliki probabilitas paling tinggi dengan menghitung frekuensi data input yang sesuai classnya misalnya data input age memiliki 3 jenis inputan yaitu young, adult, dan old. Hitung jumlah data young jika output >50K, jumlah data adult jika output >50K, data old jika output>50K, data young jika output <=50K, jumlah data adult jika output <=50K, dan data old jika output<=50K.

Setelah frekuensi didapat dihitung probabilitas tiap data terhadap class dengan cara

$P_{data|X} = f_{data|X}/n_X$

$P_{data|X}$ adalah Probabilitas data pada class X, $f_{data|X}$ merupakan frekuensi kemunculan data pada class X, dan n_x merupakan banyak data mengoutputkan class X

Setelah itu dicari probabilitas kemunculan class pada data test yang diuji. Terdapat inputan data test sebanyak 40 data. Dengan menggunakan rumus, peluang output class dapat dihitung

$$P(class = X|P_{data1|X}P_{data2|X\dots}P_{datan|X})$$

P merupakan peluang munculnya class X dengan mengkalikan probabilitas setiap input data.

Dibandingkan P(>50K) dengan P(<=50K) sehingga bila lebih besar maka output class itu lah yang menjadi output class data.

Dikarenakan ada data yang memiliki $P_{data|X}$ =0 maka digunakan laplace smoothing dengan cara menambah tiap $f_{data|X}$ dengan 1. Hal ini dilakukan agar saat penghitungan P(class) tidak samadengan 0

Strategi

Pertama import data Trainset kedalam software seperti pada lampiran gambar1. Lalu Hitung tiap frekuensi inputan seperti lampiran Gambar 2. Bagian ini akan menghitung frekuensi data train tiap input dengan class masing-masing. Dikarenakan menggunakan laplace smoothing inisial frekuensi di set 1. Setelah semua frekuensi di dapatkan dicari probabilitas data sesuai input dan class masing-masing yang akan digunakan untuk mencari output data test. Cara perhitungan dapat dilihat di Gambar3.

Setelah penggunaan data train, di import data baru berupa datatest dengan cara yang sesuai Gambar 4. Lalu dicocokan setiap inputan dan dikalikan probabilitasnya sesuai kemungkinan 2 class yang berbeda seperti yang ditulis pada lampiran Gambar 5. Setelah itu di bandingkan P(>50K) dengan P(<=50K) lalu dipilih yang memiliki probabilitas terbesar sesuai aturan pada Gambar 6.

Saat semua data test diberi output class, di buat file hasil sesuai lampiran Gambar 7.

Untuk pengujian akurasi tidak dapat dilakukan karena belum diketahui data yang benar sehingga tidak dapat di hitung.

Lampiran

```go
func main() {
    //import csv
    csvFile, _ := os.Open("TrainsetTugas1ML.csv")
    reader := csv.NewReader(bufio.NewReader(csvFile))
    var trainml []Train
    i := 0
    for {

        line, error := reader.Read()
        if error == io.EOF {
            break
        } else if error != nil {
            log.Fatal(error)
        }
        if i == 0 {
            i++
            continue
        }
        trainml = append(trainml, Train{
            Id:           line[0],
            Age:          line[1],
            Workclass:    line[2],
            Education:    line[3],
            Marital:      line[4],
            Occupation:   line[5],
            Relationship: line[6],
            Hour:         line[7],
            Income:       line[8],
        })
    }
    fmt.Println("data train telah di import")
```

Gambar1

```
//Mulai penghitungan frequency
for j := 0; j < 160; j++ {
    //freq age & class
    if trainml[j].Income == ">50K" {
        if trainml[j].Age == "young" {
            fYoungR++
        }
        if trainml[j].Age == "adult" {
            fAdultR++
        }
        if trainml[j].Age == "old" {
            fOldR++
        }
    } else {
        if trainml[j].Age == "young" {
            fYoungP++
        }
        if trainml[j].Age == "adult" {
            fAdultP++
        }
        if trainml[j].Age == "old" {
            fOldP++
        }
    }
```

Gambar 2

```
//probalitas dengan laplace
//P Age >50K
var pAdultR = fAdultR / (fAdultR + fOldR + fYoungR)
var pOldR = fOldR / (fAdultR + fOldR + fYoungR)
var pYoungR = fYoungR / (fAdultR + fOldR + fYoungR)
//P AGe <=50K
var pAdultP = fAdultP / (fAdultP + fOldP + fYoungP)
var pOldP = fOldP / (fAdultP + fOldP + fYoungP)
var pYoungP = fYoungP / (fAdultP + fOldP + fYoungP)
```

Gambar 3

```
//import TestsetTugas1ML.csv
csvFile, _ = os.Open("TestsetTugas1ML.csv")
reader = csv.NewReader(bufio.NewReader(csvFile))
var testml []Train
i = 0
for {

    line, error := reader.Read()
    if error == io.EOF {
        break
    } else if error != nil {
        log.Fatal(error)
    }
    if i == 0 {
        i++
        continue
    }
    testml = append(testml, Train{
        Id:           line[0],
        Age:          line[1],
        Workclass:    line[2],
        Education:    line[3],
        Marital:      line[4],
        Occupation:   line[5],
        Relationship: line[6],
        Hour:         line[7],
    })
}
```

Gambar 4

```
//cek Age
if testml[j].Age == "young" {
    InR = InR * pYoungR
    InP = InR * pYoungP
} else if testml[j].Age == "adult" {
    InR = InR * pAdultR
    InP = InR * pAdultP
} else {
    InR = InR * pOldR
    InP = InR * pOldP
}
```

Gambar 5

```
//kesimpulan income
if InR > InP {
    testml[j].Income = ">50K"
} else {
    testml[j].Income = "<=50K"
}
```

Gambar 6

```
file, err := os.Create("TebakanTugas1ML.csv")
checkError("Cannot create file", err)
defer file.Close()

writer := csv.NewWriter(file)
defer writer.Flush()
// for _, value := range testml {
//  err := writer.Write(string(value))
//  checkError("Cannot write to file", err)
// }
var records = [][]string{
    {"id", "age", "workclass", "education", "marital-status", "occupation", "relationship", "hours-per-wee
}
err = writer.WriteAll(records)
for j := 0; j < 40; j++ {
    records = [][]string{
        {testml[j].Id, testml[j].Age, testml[j].Workclass, testml[j].Education, testml[j].Marital, testml[
    }
    err = writer.WriteAll(records)
    checkError("Cannot write to file", err)
}
fmt.Println("data sudah di export. Eksekusi selesai")
```

Gambar 7