**Pokok Bahasan VIII**

**Naïve Bayes dengan Python**

**Kode Pokok Bahasan**: TIK.RPL03.005.00.01

**Deskripsi Pokok Bahasan**:

Membahas bagaimana penerapan Algoritma Naïve Bayes pada dataset titanic.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Elemen Kompetensi | Indikator Kinerja | Jml Jam | Hal |
| 1. | Menampilkan peluang dari kasus yang diberikan. | Mampu melakukan analisis terhadap peluang atas kejadian yang ditentukan. | 1 | 12 |
| 2. | Melakukan perhitungan menggunakan R. | Mampu melakukan perhitungan peluang menggunakan R pada kondisi yang ditentukan | 2 | 15 |
| 3. | Menggunakan fungsi naiveBayes dalam memprediksi data | Mengimplementasikan fungsi naive bayes pada prediksi data |  |  |

**TUGAS PENDAHULUAN**

Hal yang harus dilakukan dan acuan yang harus dibaca sebelum praktikum :

1. Menginstal R pada PC masing-masing praktikan.

2. Menginstal R Studio pada PC masing-masing praktikan.

**DAFTAR PERTANYAAN**

1. Apa itu algoritma Naïve Bayes?

Algoritma Naive Bayes adalah algoritma yang mempelajari probabilitas suatu objek dengan ciri-ciri tertentu yang termasuk dalam kelompok/kelas tertentu. Singkatnya, ini adalah pengklasifikasi probabilistik.

2. Apa kegunaan Naïve Bayes?

Prediksi multi-kelas: Algoritma klasifikasi Naive Bayes dapat digunakan untuk memprediksi probabilitas posterior dari beberapa kelas variabel target. Klasifikasi teks: Karena fitur prediksi multi-kelas, Naive Bayes algoritma klasifikasi sangat cocok untuk klasifikasi teks

3. Sebutkan tahapan dari proses algoritma Naïve Bayes!

Menghitung jumlah kelas/label. Menghitung jumlah kasus perkelas. Mengalikan semua hasil variable kelas. Membandingkan hasil perkelas.

**TEORI SINGKAT**

Algoritma Naive Bayes merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik yg dikemukakan oleh ilmuwan Inggris Thomas Bayes. Algoritma Naive Bayes memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dr Naïve Bayes Classifier ini adalah asumsi yg sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian.

Naive Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini dibuktikan pada jurnal Xhemali, Daniela, Chris J. Hinde, and Roger G. Stone. “Naive Bayes vs. decision trees vs. neural networks in the classification of training web pages.” (2009), mengatakan bahwa “Naïve Bayes Classifier memiliki tingkat akurasi yg lebih baik dibanding model classifier lainnya”.

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

**LAB SETUP**

Hal yang harus disiapkan dan dilakukan oleh praktikan untuk menjalankan praktikum modul ini.

1. Menginstall library yang dibutuhkan untuk mengerjakan modul.

2. Menjalankan Jupyter Notebook.

**ELEMEN KOMPETENSI I**

**Deskripsi:**

Menampilkan peluang dari kasus yang diberikan.

**Kompetensi Dasar**:

Mampu melakukan analisis terhadap peluang atas kejadian yang ditentukan.

**Latihan**

**Penjelasan Singkat :**

Pada latihan ini anda akan diminta untuk menampilkan summarize dari data menggunakan python.

**Langkah-Langkah Praktikum:**

1.Instal package berikut

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

2. Buat dataframe yang digunakan

df=pd.read\_csv("train\_clean.csv")

df\_test=pd.read\_csv("test\_clean.csv")

3. Membuat sampel train dan test

train, test = train\_test\_split(df, test\_size=0.2)

4. Jalankan perintah berikut

Hitung probabilitas selamat penumpang Titanic yang berjenis kelamin perempuan.

surviving\_female = df[(df['Sex'] == "female") & (df['Survived'] == 1)]

P\_yes=len(surviving\_female)/len(train)

P\_yes

Output :

|  |
| --- |
|  |

**ELEMEN KOMPETENSI II**

**Deskripsi:**

Melakukan perhitungan manual menggunakan R.

**Kompetensi Dasar**:

Mampu melakukan perhitungan peluang secara manual menggunakan R pada kondisi yang ditentukan

**Latihan 1.2.1**

**Penjelasan Singkat :**

Pada latihan ini anda akan diminta untuk melakukan perhitungan peluang secara manual menggunakan R.

**Langkah-Langkah Praktikum:**

Lakukan perhitungan kasus di atas dengan menggunakan R.

Output :

|  |
| --- |
|  |

**ELEMEN KOMPETENSI III**

**Deskripsi:**

Menggunakan fungsi naïve bayes dalam memprediksi data.

**Kompetensi Dasar**:

Mengimplementasikan fungsi naive bayes pada prediksi data

**Latihan 1.3.1**

**Penjelasan Singkat :**

Pada latihan ini anda akan diminta untuk mengimplementasikan naïve bayes pada kasus yang diberikan.

**Langkah-Langkah Praktikum:**

Gunakan [titanic.csv](http://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv) yang berisi data 887 penumpang Titanic passengers. Kolom data menggambarkan survived (*S*), age (*A*), passenger-class (*C*), sex (*G*) and the fare paid (*X*).

Hitung peluang bersyarat (conditional probability) di bawah ini

*P*(*S*= true | *G*=female)

*P*(*S*= true | *G*=male)

*P*(*S*= true | *C*=1)

*P*(*S*= true | *C*=2)

*P*(*S*= true | *C*=3)

*P*(*S*= true | *G*=female,*C*=1) =

*P*(*S*= true | *G*=female,*C*=2) =  
*P*(*S*= true | *G*=female,*C*=3) =  
*P*(*S*= true | *G*=male,*C*=1) =  
*P*(*S*= true | *G*=male,*C*=2) =  
*P*(*S*= true | *G*=male,*C*=3) =

Gunakan Python dan R :

|  |
| --- |
|  |

Jalankan perintah R di bawah ini :

|  |
| --- |
| import pandas as pd  import numpy as np  titanic = pd.read\_csv("train\_clean.csv")  titanic.head() |

Output :

|  |
| --- |
|  |

|  |
| --- |
| titanic.drop(['Name','SibSp','Parch','Ticket','Cabin','Embarked'],axis='columns',inplace=True)  titanic.head() |

Output :

|  |
| --- |
|  |

|  |
| --- |
| # Variabel independen  x = titanic.drop(["Survived"], axis = 1)  x.head()  # Variabel dependen  y = titanic["Survived"]  y.head() |

Output :

|  |
| --- |
|  |

|  |
| --- |
| x['Sex'].replace(['female','male'],[0,1],inplace=True)  x.head() |

Output :

|  |
| --- |
|  |

|  |
| --- |
| x.Age = x.Age.fillna(x.Age.mean())  x.head() |

Output :

|  |
| --- |
|  |

|  |
| --- |
| from sklearn.model\_selection import train\_test\_split  x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 123)  from sklearn.naive\_bayes import GaussianNB  # Mengaktifkan/memanggil/membuat fungsi klasifikasi Naive Bayes  modelnb = GaussianNB()  # Memasukkan data training pada fungsi klasifikasi Naive Bayes  nbtrain = modelnb.fit(x\_train, y\_train)  y\_pred = nbtrain.predict(x\_test)  y\_pred |

Output :

|  |
| --- |
|  |

|  |
| --- |
| np.array(y\_test) |

Output :

|  |
| --- |
|  |

|  |
| --- |
| nbtrain.predict\_proba(x\_test) |

Output :

|  |
| --- |
|  |

|  |
| --- |
| from sklearn.metrics import confusion\_matrix  confusion\_matrix(y\_test, y\_pred) |

Output :

|  |
| --- |
|  |

Berikan penjelasan terhadap output di atas

|  |
| --- |
|  |

**CEK LIST**

|  |  |  |  |
| --- | --- | --- | --- |
| Elemen Kompetensi | No Latihan | Penyelesaian | |
| Selesai | Tidak selesai |
| 1 | 1.1.1 |  |  |
| 2 | 1.2.1 |  |  |
| 3 | 1.3.1 |  |  |

**FORM UMPAN BALIK**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Elemen Kompetensi** | **Tingkat Kesulitan** | | | **Tingkat Ketertarikan** | | | **Waktu Penyelesaian dalam menit** |
| Memahami data pre-processing. |  |  |  |  |  |  |  |
|  |  | Sangat Mudah |  |  | Tidak Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Mudah |  |  | Cukup Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | * Biasa |  |  | Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Sulit |  |  | * Sangat Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Sangat Sulit |  |  |  |  |
|  |  |  |  |  |  |  |
| Mengimplementasikan pre-processing data. |  |  |  |  |  |  |  |
|  |  | Sangat Mudah |  |  | Tidak Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Mudah |  |  | Cukup Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | * Biasa |  |  | Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Sulit |  |  | * Sangat Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Sangat Sulit |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Elemen Kompetensi** | **Tingkat Kesulitan** | | | **Tingkat Ketertarikan** | | | **Waktu Penyelesaian dalam menit** |
| Menggunakan fungsi naiveBayes dalam memprediksi data |  |  |  |  |  |  |  |
|  |  | Sangat Mudah |  |  | Tidak Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Mudah |  |  | Cukup Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | * Biasa |  |  | Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Sulit |  |  | * Sangat Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Sangat Sulit |  |  |  |  |
|  |  |  |  |  |  |  |