Bab 4
Naive Bayes

A. KOMPETENSI DASAR

- Memahami konsep *Naive Bayes*.
- Memahami eksperimen klasifikasi menggunakan Naive Bayes
- Memahami membuat model Naïve Bayes

B. ALOKASI WAKTU

4 js (4x50 menit)

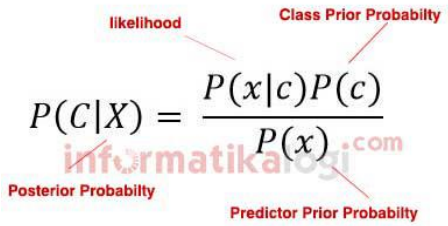
C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami Tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas dengan baik, sabar, dan jujur.
- Tanyakan kepada teman lalu asisten/dosen apabila ada hal-hal yang kurang jelas.

D. DASAR TEORI HIERARCHICAL CLUSTERING

Video singkat tentang Naive Bayes: https://www.youtube.com/watch?v=CPq0CI0ahss

Naive Bayes adalah model machine learning yang dapat digunakan dalam berbagai tugas klasifikasi. Aplikasi yang umum termasuk memfilter spam, mengklasifikasikan dokumen, prediksi sentimen, dll. Teknik klasifikasi model ini sangat cepat dan sederhana yang sering cocok untuk dataset berdimensi sangat tinggi dikarenakan memiliki sedikit parameter yang bisa diubah, maka akhirnya menjadi sangat berguna sebagai dasar klasifikasi. Naive Bayes menggunakan teorema Bayes, yang merupakan persamaan yang menggambarkan hubungan probabilitas bersyarat jumlah statistik. Berikut ini persamaannya:



Catatan!

X : Vektor input ***c*** : Sebuah class spesifik ***P(C|X)*** : Probabilitas class berdasar vektor input yang diketahui (posteriori probability) ***P(c)*** : Probabilitas class yang dicari (prior probability) ***P(X|C)*** : Probabilitas tiap input berdasarkan kondisi pada class ***P(c)*** : Probabilitas sebuah class dari keseluruhan class yang ada

Features atau X terdapat:

$$X = (x_1, x_2, x_3, ..., x_n)$$

Yang dimana x1, x2, x3….xn merupakan jumlah fitur yang dimiliki pada datasets. Sehingga, persamaan dapat dibentuk seperti ini:

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Berikut contoh penerapan Naive Bayes yaitu terdapat sebuah datasets berisi kondisi cuaca dan hasil klasifikasinya berupa “yes” atau “no” untuk bermain di luar rumah.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Hal yang dilakukan untuk menerapkan model ini, konversikan data set berbentuk tabel frekuensi dan mencari probabilitas seperti peluang Overcast = 0.29 dan peluang untuk bermain = 0.64

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Sekarang, gunakan persamaan Naive Bayesian untuk menghitung probabilitas posterior untuk setiap kelas. Kelas dengan probabilitas posterior tertinggi adalah *outcome prediction*. Misal, pemain bermain dilapangan ketika cuacanya cerah, maka hasil probabilitas posterior nya sebesar 0.60. Cara perhitungan sebagai berikut:

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)
P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64
P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60

Pada kasus tersebut, variabel kelas (y) hanya memiliki dua hasil, "**yes**" atau "**no"**. Mungkin ada kasus di mana klasifikasi bisa multivarian. Oleh karena itu, diperlukan untuk menemukan kelas y dengan probabilitas maksimum.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Scikit-learn menyediakan algoritma Naïve Bayes dengan memiliki macam-macam teknik klasifikasinya terdapat Gaussian, Multinominal, Bernoulli, dan lain-lain. Diantara teknik tersebut, terdapat cara perhitungan mencari probabilitas terhadap suatu kondisi atau fitur pada datasets. Berikut persamaan menggunakan Gaussian Naïve Bayes:

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

Yang dimana σ_y dan μ_y telah diestimasikan menggunakan "maximum likelihood". Teknik Gaussian ini digunakan saat label dataset tersebut adalah *continuous value*, bukan *discrete value* (1 or 0).

Catatan! Link untuk perhitungan *conditional probability* menggunakan Gaussian, Multinomial, Complement dan Bernoulli
https://scikit-learn.org/stable/modules/naive_bayes.html

KELEBIHAN NAIVE BAYES:

1. Mudah dan cepat dalam memprediksi sebuah class berdasarkan input
2. Memiliki performa lebih baik dibandingkan dengan classifier yang lain, JIKA, asumsi bahwa semua variabel input adalah independen (tidak saling mempengaruhi) terhadap satu sama lain.
3. Cocok untuk dataset dengan input variabel ber-tipe kategorikal atau nominal
4. Prediktor yang baik walaupun dengan dataset yang kecil

KEKURANGAN NAIVE BAYES:

1. Jika input variabel yang dicari tidak memiliki observasi pada training dataset maka menyebabkan nilai 0 untuk perhitungannya dan prediksi gagal. Solusinya dapat menggunakan Laplace Estimation atau mengganti observasi bernilai 0 dengan 1.
2. Terkenal sebagai prediktor yang buruk, karena hasil prediksi adalah asumsi
3. Asumsi tentang independensi antara input variabel sangatlah "*naive*" dan tidak berlaku dalam dataset dari kehidupan nyata.

E. LATIHAN (jawaban dapat ditulis pada halaman baru)

1. Persiapkan data berupa array 2 dimensi sebagai berikut: `np.array([[7,3], [1,10], [5,5], [2,1], [3,3], [2,7], [17,14], [6,8], [7,5], [3,6]])` sebagai variable X. Kemudian buatkan array 1 dimensi: `np.array([1, 1, 1, 2, 2, 2, 1, 2, 1])`
2. Implementasikan Naïve Bayes classifier:

```
from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()
clf.fit(X, Y)
print(clf.predict([[-0.8, -1]]))
clf_pf = GaussianNB()
clf_pf.partial_fit(X, Y, np.unique(Y))
print(clf_pf.predict([[-0.8, -1]]))
```

3. Buatlah dataset make_blobs dan visualisasikan

```
X, y = make_blob(100, 2, center=2, random_state=2, cluster_std=1.5)
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='RdBu');
model = GaussianNB()
modul.fit(X, y);
Generate X dan Y baru di variable lain
rng = np.random.RandomState(0)
Xnew = [-6, -14] + [14, 18] * range.rand(2000, 2)
ynew = model.predict(Xbaru)
Visualisasikan hasil prediksi
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='RdBu')
#lim = plt.axis()
plt.scatter(Xbaru[:, 0], Xbaru[:, 1], c=ynew, s=20, cmap='RdBu', alpha=0.1)
plt.axis(lim);
```

F. TUGAS

1. Apa perbedaan antara metode “partial_fit” dengan “fit”
2. Carilah datasets (ukuran kecil) untuk mengimplementasikan klasifikasi Naïve Bayes!
3. Prediksi hasil test data menggunakan Naïve Bayes tanpa menggunakan fungsi model yang disediakan oleh python, pilih salah satu teknik perhitungan “condition probability” (Multinominal, Gaussian, atau Bernoulli) yang cocok dengan jenis dataset anda.
4. Bandingkan hasil prediksi Naïve yang anda buat dengan yang disediakan oleh library sklearn menggunakan accuracy_score dan visualisasi data seperti di Latihan.
5. Buatlah kesimpulan tentang Model Naïve Bayes anda.