BINUS University

Academic Career: ***~~Undergraduate / Master / Doctoral /~~ BINUS Online ~~/ Professional~~**)*	**Class Program:** *Regular ~~/ Global Class~~*)*
☐ **Mid Exam** ☐ **Others Exam :**_____ ■ **Final Exam**	**Term : Odd ~~/ Even / Compact~~*)** **Period (Only for *BINUS Online ~~/ Master~~*) : ~~1 /~~ 2*)**
☐ **Kemanggisan** ☐ **Senayan** ☐ **Semarang** ☐ **Alam Sutera** ☐ **Bandung** ☐ **Medan** ☐ **Bekasi** ☐ **Malang** ■ **BiOn**	**Academic Year :** **2025 / 2026**
Exam Type* : ~~Onsite / Online /~~ Take Home	Faculty / Dept. : BINUS Online Learning
Day / Date** : Senin, 09 Februari 2026 s.d. Senin, 16 Februari 2026	Code - Course : COMP6936036 - Machine Learning
Time** : 00:00 WIB s.d. 12:00 WIB (Siang)	
Exam Specification*** : ■ Open Book ■ Open Notes ☐ Close Book ☐ Oral Test ■ Open E-Book	Class : All Classes
Equipment*** : ☐ Examination Booklet ■ Laptop ☐ Drawing Paper - A3 ■ Calculator ☐ Tablet ☐ Drawing Paper - A2 ☐ Dictionary ☐ Smartphone ☐ Notes :_____sheet	Student ID*** : 2802536146 Name*** : Asraf Muhammad Izzuddin Signature*** : 

) Strikethrough the unnecessary items* *) For Online Exam, this is the due date* ****) Only for Onsite Exam*

Please insert the test paper into the examination booklet and submit both papers after the test.***
The penalty for CHEATING is DROP OUT!

INSTRUCTION

Kerjakan soal dengan baik, bacalah petunjuk pengerjaan soal dengan teliti dan tuntas. Segala macam bentuk pelanggaran akan dikenakan sangsi sesuai dengan aturan yang berlaku.

1. LO 1: the concept of methods in machine learning (20%)

Para peneliti sedang membuat model untuk klasifikasi penyakit pasien berdasakan data genetik. Data genetik ini terdiri atas 200 samples dengan 10000 ekrpresi gen yang merupakan rangkaian proses penggunaan informasi dari suatu gen.

a. Bagaimana curse of dimensionality mempengaruhi model non-parametric dalam kasus ini?

Jawab:

Sebelum menjawab pertanyaan itu, perlu diketahui terlebih dahulu mengenai apa aitu curse of dimensionality. Curse of dimensionality adalah kondisi ketika jumlah fitur atau variabel dalam suatu data jauh lebih banyak dibandingkan jumlah sampelnya. Dalam kasus soal No. 1, terdapat 200 samples/pasien tetapi masing-masing memiliki 10.000 ekspresi gen sebagai fitur. Artinya, model harus bekerja dalam ruang dengan 10000 dimensi hanya berdasarkan 200 titik data. Kondisi ini membuat data terasa “terlalu menyebar” sehingga sulit untuk melihat pola yang jelas di dalamnya. Selain itu, ketika jumlah dimensinya sangat banyak, perbedaan jarak antar data menjadi tidak terlalu berarti karena hampir semua data terlihat sama-sama jauh satu sama lain. Akibatnya, model menjadi kesulitan untuk mengenali pola yang benar-benar konsisten dan sulit membuat prediksi yang bisa berlaku dengan baik pada data baru.

Masalah ini menjadi lebih serius ketika menggunakan model non-parametric seperti Decision Tree atau KNN. Model non-parametric bersifat fleksibel dan tidak membatasi bentuk hubungan antar variabel, sehingga dapat terus menambah kompleksitas sesuai data yang ada. Namun, dalam kondisi fitur sangat banyak dan sampel sedikit, fleksibilitas ini justru membuat model mudah menghafal noise atau pola kebetulan pada data training, yang dikenal sebagai overfitting. Model mungkin terlihat sangat akurat saat diuji pada data training, tetapi performanya menurun drastis ketika diuji pada data baru.

b. Menurut Anda mengapa model parametric dengan regularisasi yang kuat lebih dipilih untuk data tersebut?

Jawab:

Model parametric seperti Logistic Regression atau Linear SVM memiliki struktur yang lebih sederhana dan terkontrol. Model ini mengasumsikan bentuk hubungan tertentu, misalnya kombinasi linear antar fitur, sehingga kompleksitasnya lebih terbatas. Struktur yang lebih disiplin ini membuat model lebih stabil ketika jumlah data sedikit. Namun, karena jumlah fitur sangat besar, model parametric tetap perlu dibantu dengan regularisasi. Regularisasi adalah teknik yang memberikan penalti pada koefisien yang terlalu besar agar model tidak menjadi terlalu kompleks.

Dalam konteks data genetik, regularisasi seperti L1 atau Lasso sangat berguna karena selain mengontrol kompleksitas, metode ini juga dapat melakukan feature selection secara otomatis. Banyak koefisien gen yang tidak relevan akan ditekan menjadi nol, sehingga hanya gen yang benar-benar berkontribusi terhadap prediksi yang dipertahankan. Hal ini sangat penting karena dari 10000 gen, biasanya hanya sebagian kecil yang benar-benar berhubungan dengan penyakit. Dengan regularisasi, model menjadi lebih sederhana, lebih stabil, dan lebih mudah diinterpretasikan secara biologis.

Secara matematis, ketika jumlah fitur jauh lebih besar dari jumlah sampel, estimasi parameter menjadi tidak stabil dan varians model menjadi sangat tinggi. Regularisasi membantu menurunkan varians tersebut dengan sedikit menambah bias, sehingga tercapai keseimbangan yang lebih baik antara bias dan varians. Hasil akhirnya adalah model yang memiliki kemampuan generalisasi lebih baik pada data baru. Oleh karena itu, dalam kasus 200 samples dan 10000 gen, pendekatan yang paling rasional adalah menggunakan model parametric dengan regularisasi kuat, seperti Logistic Regression dengan L1 atau Linear SVM dengan penalti reguler, sebelum mempertimbangkan model yang lebih kompleks.

c. Risiko apa yang secara teoritis akan muncul jika model non-parametric digunakan tanpa dimensionality reduction?

Jawab:

Secara teoritis, dari penjelasan jawaban No. 1.a. pada paragraph dua, jika model non-parametric digunakan langsung pada data 200 samples dengan 10000 fitur tanpa melakukan dimensionality reduction, beberapa risiko akan muncul, diantaranya adalah:

1. Pertama adalah overfitting ekstrem. Model non-parametric seperti Decision Tree, Random Forest, atau KNN sangat fleksibel dan dapat menyesuaikan diri hampir tanpa batas terhadap data training. Dalam kondisi fitur sangat banyak dan sampel sedikit, model akan dengan mudah menemukan pola yang sebenarnya hanya kebetulan atau noise. Secara teori, ketika dimensi meningkat, kompleksitas fungsi yang dapat dibentuk model juga meningkat drastis, sehingga model lebih cenderung menghafal data daripada belajar pola umum. Akibatnya, error pada data training sangat kecil, tetapi error pada data baru menjadi besar.

2. Risiko kedua adalah masalah yang disebut sparsity, yaitu data menjadi terasa sangat “jarang” ketika jumlah dimensinya sangat banyak. Di kasus yang menempatkan 200 titik data di ruang dengan 10000 arah yang berbeda. Ruang tersebut menjadi sangat luas, sementara jumlah titiknya tetap sedikit. Akibatnya, setiap titik data terlihat sangat berjauhan satu sama lain. Karena jaraknya menjadi hampir sama-sama jauh, metode yang mengandalkan kedekatan antar data, seperti KNN atau pemisahan pada decision tree, menjadi kurang efektif. Konsep “tetangga terdekat” tidak lagi jelas, karena hampir semua data tampak sama jauhnya. Hal ini membuat model kesulitan menentukan pola yang benar-benar bermakna dan akhirnya menjadi tidak stabil

3. Risiko ketiga adalah inefisiensi komputasi. Model non-parametric dalam dimensi sangat tinggi membutuhkan komputasi besar, baik dalam pencarian split terbaik pada tree maupun dalam perhitungan jarak pada KNN. Ini memperburuk stabilitas dan memperlambat proses training.

d. Bagaimana peningkatan volume data akan mempengaruhi pilihan model yang digunakan?

Jawab:

Pada kasus ini, masalah utamanya adalah jumlah fitur jauh lebih besar dibanding jumlah data atau bisa disebut juga dengan “Big-p, Little-n”. Karena itu sebelumnya memilih model parametric dengan regularisasi kuat merupakan pilihan yang lebih tepat agar model tetap stabil dan tidak overfitting.

Sekarang apabila jumlah pasien meningkat, misalnya menjadi 2000, 5000, atau bahkan 20000 pasien. Peningkatan volume data ini akan sangat memengaruhi pilihan model. Berikut penjabarannya:

1. Ketika jumlah sampel bertambah, ruang data menjadi lebih “terisi”. Data tidak lagi terlalu jarang dalam ruang 10000 dimensi. Dengan lebih banyak contoh, model memiliki informasi yang cukup untuk membedakan pola yang benar-benar konsisten dari noise. Risiko overfitting berkurang karena pola yang muncul berulang kali di banyak pasien lebih mungkin mencerminkan hubungan biologis yang nyata.

2. Kedua, model non-parametric menjadi lebih layak digunakan. Dengan data yang lebih banyak, fleksibilitas model seperti Random Forest, Gradient Boosting, atau bahkan model yang lebih kompleks dapat dimanfaatkan tanpa terlalu khawatir model hanya menghafal data.

3. Ketiga, kebutuhan regularisasi mungkin tetap ada, tetapi tidak perlu sekuat sebelumnya. Ketika data sangat sedikit, regularisasi kuat diperlukan untuk mencegah model menjadi liar. Namun ketika data cukup banyak, model bisa belajar struktur yang lebih kompleks secara stabil. Bahkan pendekatan non-linear bisa memberikan performa lebih baik dibanding model linear sederhana.

Namun, meskipun volume data meningkat, dimensionality reduction atau feature selection tetap sering digunakan dalam data genetik. Ini bukan hanya untuk mencegah overfitting, tetapi juga untuk meningkatkan interpretabilitas dan efisiensi komputasi.

Sederhananya, semakin banyak data tersedia, semakin fleksibel model yang bisa digunakan. Pada 200 pasien, model parametric dengan regularisasi kuat adalah pilihan aman. Namun, ketika jumlah pasien meningkat signifikan, model non-parametric dan model yang lebih kompleks menjadi lebih realistis dan potensial memberikan akurasi yang lebih tinggi.

2. LO 2: a model problem using machine learning methods (20%)

Suatu rumah sakit ingin membangun sistem pendukung diagnosis medis dengan menggunakan model Support Vector Machine. Data dari 2500 pasien terdiri atas umur, tekanan darah, tingkat kolesterol, status perokok (Y/N), riwayat keluarga (Y/N) dan keberadaan penyakit (Y/N).

a. Tentukan tipe dari tiap fitur dan target dari data tersebut.

Jawab:

Pada kasus tersebut, umur, tekanan darah, dan tingkat kolesterol adalah fitur numerik kontinu karena nilainya berbentuk angka dan dapat berada pada rentang tertentu. Sementara status perokok (Y/N) dan riwayat keluarga (Y/N) adalah fitur kategorikal biner karena hanya memiliki dua nilai kategori (Ya atau Tidak). Adapun keberadaan penyakit (Y/N) adalah target sekaligus variabel kategorikal biner yang ingin diprediksi oleh model, yaitu apakah pasien memiliki penyakit atau tidak.

b. Bagaimana model support vector machine digunakan untuk menyelesaikan masalah non-linear yang mungkin timbul pada data medis tersebut? Jelaskan secara matematis.

Jawab:

Model SVM menyelesaikan masalah klasifikasi dengan mencari hyperplane yang memisahkan dua kelas dengan margin terbesar. Jika data bisa dipisahkan secara linear, bentuk hyperplane-nya dapat dituliskan sebagai

$$w^\top x + b = 0$$

dan keputusan kelas dilakukan dengan tanda dari fungsi Keputusan

$$f(x) = \text{sign}(w^\top x + b).$$

Namun pada data medis, hubungan antar fitur sering non-linear. Untuk mengatasi ini, SVM menggunakan kernel trick, yaitu memetakan data dari ruang asli ke ruang fitur berdimensi lebih tinggi melalui fungsi

$$\phi(x)$$

lalu membangun pemisah linear di ruang tersebut. Secara matematis, SVM tidak perlu menghitung $\phi(x)$secara eksplisit karena cukup menggunakan kernel

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle.$$

Fungsi keputusan SVM kernel menjadi

$$f(x) = \text{sign}(\sum_{i=1}^{n} \alpha_i \, y_i K(x_i, x) + b)$$

dengan $\alpha_i$adalah bobot yang dipelajari dari data dan hanya sebagian data (support vectors) yang berkontribusi besar pada pemisah. Dengan cara ini, pola non-linear seperti interaksi umur. kolesterol. dan tekanan darah yang kompleks dapat tetap dipisahkan secara efektif.

c. Bagaimana problem imbalance akan mempengaruhi model SVM?

Jawab:

Problem imbalance terjadi ketika jumlah pasien “penyakit = Y” jauh lebih sedikit atau jauh lebih banyak daripada “penyakit = N”. Pada SVM, kondisi ini dapat membuat model cenderung memilih hyperplane yang menguntungkan kelas mayoritas karena secara optimasi, kesalahan pada kelas minoritas “terlihat” lebih kecil dampaknya terhadap total error. Akibatnya, model bisa tampak akurat secara keseluruhan tetapi gagal mengenali pasien yang benar-benar sakit. Pada SVM, penanganan umum adalah memberi bobot kelas (class weight) lebih besar untuk kelas minoritas, melakukan oversampling/undersampling, atau mengevaluasi dengan metrik yang tidak bias terhadap kelas mayoritas.

d. Risiko etika apa yang muncul akibat kesalahan klasifikasi dengan model SVM?

Jawab:

Risiko etika paling besar dari kesalahan klasifikasi pada sistem diagnosis adalah dampaknya langsung pada keselamatan pasien. Jika terjadi false negative (pasien sebenarnya sakit tetapi diprediksi tidak sakit), pasien bisa terlambat mendapatkan perawatan, penyakit memburuk, bahkan berisiko fatal. Jika terjadi false positive (pasien sebenarnya tidak sakit tetapi diprediksi sakit), pasien bisa mengalami kecemasan, menerima pemeriksaan lanjutan yang tidak perlu, atau bahkan mendapat terapi yang sebenarnya tidak dibutuhkan. Selain itu, bisa terjadi model lebih sering salah pada kelompok tertentu jika data latih tidak representatif, sehingga muncul potensi diskriminasi layanan kesehatan. Oleh karena itu, sistem seperti ini harus disertai evaluasi yang tepat (misalnya fokus pada sensitivitas/recall untuk kelas penyakit), pengujian ketat pada data baru, serta mekanisme human-in-the-loop agar keputusan akhir tetap melibatkan tenaga medis.

3. LO 2: a model problem using machine learning methods (20%)

Seorang pemain menarik dua tuas yang memiliki reward Bernoulli, model dari peristiwa acak tunggal dengan dua kemungkinan hasil. Pemain telah melakukan 5 kali penarikan tuas dengan hasil berikut:

Tuas A: 0 keberhasilan

Tuas B: 3 keberhasilan

Suatu agent harus memutuskan mesin mana yang akan ditarik tuasnya dari waktu ke waktu.

a. Mengapa greedy policy sederhana akan memilih opsi yang kurang optimal?

Jawab:

Greedy policy berarti selalu memilih tuas dengan rata-rata hasil tertinggi berdasarkan data yang sudah ada. Karena data awal di kasus No. 3 menunjukkan B punya 3 sukses dan A punya 0 sukses, greedy hampir pasti akan memilih B terus menerus. Namun, masalahnya data awal sangat sedikit, jadi rata-rata itu bisa bias dan belum mencerminkan kualitas sebenarnya. Ada dua alasan utama:

1. Sampel kecil membuat estimasi tidak stabil

   Kalau A baru ditarik sedikit, lalu kebetulan hasilnya 0 semua, greedy akan menganggap A buruk, padahal bisa saja A sebenarnya bagus tapi "lagi apes" di percobaan awal.

2. Greedy bisa terjebak

   Begitu greedy "terlanjur percaya" lebih baik, ia akan jarang atau bahkan tidak pernah mencoba A lagi. Akibatnya agent tidak pernah punya kesempatan menemukan bahwa A mungkin sebenarnya lebih menguntungkan.

b. Tunjukan dengan epsilon greedy dapat memperbaiki bias dari percobaan inisial

Jawab:

Epsilon greedy memperbaiki masalah di atas karena dia membagi keputusan jadi dua mode:

1. Eksploitasi dengan peluang $1 - \varepsilon$: Pilih tuas yang saat ini terlihat terbaik (misalnya B).
2. Eksplorasi dengan peluang ε: Pilih tuas secara acak untuk "coba-coba" dan menambah informasi.

Misal $\varepsilon = 0.1$ (10% eksplorasi), walaupun B terlihat lebih baik, agent masih akan mencoba A secara acak di sebagian langkah. Setiap kali A dicoba, estimasi peluang sukses A akan makin akurat. Jika ternyata A mulai menghasilkan sukses, rata-rata A naik, dan akhirnya greedy-part (90% eksploitasi) bisa beralih memilih A. Ini bisa mengurangi bias awal karena bias awal muncul akibat informasi yang kurang. Eksplorasi memaksa agent untuk tetap mengumpulkan data dari pilihan yang "sementara terlihat jelek", sehingga keputusan akhir lebih berdasarkan bukti yang cukup, bukan kebetulan di awal.

c. Asumsi waktu eksperimen=T, dengan epsilon konstant dan optimum tuas rata-rata=m. Berapa probabilitas dari optimum tuas akan dipilih dengan epsilon=0.2 dan 4 tuas?

Jawab:

Dalam ε-greedy, peluang memilih tuas optimal terdiri dari dua bagian:

1. Saat eksploitasi (peluang $1 - \varepsilon$): Agent memilih tuas terbaik menurut estimasi saat ini. Kalau diasumsikan "tuas optimal memang dipilih saat eksploitasi" (artinya estimasi sudah benar), maka peluang memilih optimal dari bagian ini adalah $1 - \varepsilon$.
2. Saat eksplorasi (peluang ε): Agent memilih acak dari 4 tuas, jadi peluang memilih tuas optimal adalah $\frac{1}{4}$.

Maka:

$$P(\text{pilih optimal}) = (1 - \varepsilon) + \varepsilon \cdot \frac{1}{4}$$

Dengan $\varepsilon = 0.2$ dan 4 tuas:

$$P = 0.8 + 0.2 \cdot 0.25 = 0.8 + 0.05 = 0.85$$

Jadi probabilitas tuas optimal dipilih adalah 85%, dengan catatan asumsi bahwa pada fase eksploitasi agent memang memilih tuas optimal.

d. Jika epsilon=0.1 dan T=1000, berapa jumlah langkah eksplorasi yang diharapkan?

Jawab:

Eksplorasi terjadi dengan peluang εpada setiap langkah. Jadi ekspektasi jumlah eksplorasi selama T langkah adalah:

$$E[\text{jumlah eksplorasi}] = \varepsilon \cdot T$$

Dengan $\varepsilon = 0.1$ dan $T = 1000$:

$$E = 0.1 \cdot 1000 = 100$$

Jadi jumlah langkah eksplorasi yang diharapkan adalah 100 langkah.

4. LO 2: a model problem using machine learning methods (20%)

Sebuah universitas ingin memprediksi apakah mahasiswa akan Lulus (1) atau Gagal (0) dalam ujian berdasarkan:

- Jam belajar per hari (x_1)

- Presentase kehadiran (x_2)

ID	Study hours	Attendancee %	Pass
1	1.0	55	0
2	2.0	60	0
3	3.0	62	0
4	3.5	65	0
5	4.0	68	0
6	6.0	75	1
7	6.5	78	1
8	7.0	80	1
9	7.5	82	1
10	8.0	85	1
11	8.5	88	1
12	9.0	90	1

Anda diminta untuk menggunakan model regresi logistik untuk memprediksi kelulusan mahasiswa

a. Buktikan bahwa fungsi sigmoid memetakan setiap input bilangan riil ke interval (0, 1).

Jawab:

Fungsi sigmoid didefinisikan sebagai:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Karena untuk setiap bilangan riil z, nilai e^{-z}selalu positif, maka penyebut $1 + e^{-z}$pasti lebih besar dari 1. Akibatnya, nilai sigmoid selalu berada di antara 0 dan 1. Untuk membuktikannya lebih konkret, bisa dilakukan substitusi beberapa nilai.

1. Contoh 1: z = 0

$$\sigma(0) = \frac{1}{1 + e^{0}} = \frac{1}{1 + 1} = \frac{1}{2} = 0{,}5$$

Nilainya berada di antara 0 dan 1.

2. Contoh 2: z = 5 (nilai positif besar)

$$\sigma(5) = \frac{1}{1 + e^{-5}}$$

Karena $e^{-5} \approx 0{,}0067$, maka:

$$\sigma(5) = \frac{1}{1 + 0{,}0067} = \frac{1}{1{,}0067} \approx 0{,}993$$

Nilainya mendekati 1.

3. Contoh 3: z = -5 (nilai negatif besar)

$$\sigma(-5) = \frac{1}{1+e^5}$$

Karena $e^5 \approx 148{,}41$, maka:

$$\sigma(-5) = \frac{1}{1+148{,}41} = \frac{1}{149{,}41} \approx 0{,}0067$$

Nilainya mendekati 0.

Dari substitusi tersebut terlihat bahwa untuk nilai z berapa pun, hasil sigmoid selalu berada di antara 0 dan 1. Ketika z besar dan positif, output mendekati 1. Ketika z besar dan negatif, output mendekati 0. Dengan demikian, fungsi sigmoid secara matematis memang memetakan seluruh bilangan riil ke dalam interval (0,1), sehingga sangat sesuai digunakan sebagai fungsi probabilitas dalam regresi logistik.

b. Jelaskan mengapa regresi logistik menghasilkan batas keputusan linier dalam feature space.

Jawab:

Pada regresi logistik, nilai probabilitas kelulusan diperoleh dari fungsi sigmoid yang diterapkan pada kombinasi linier variabel input, yaitu $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2$. Walaupun fungsi sigmoid berbentuk non-linier, proses klasifikasi dilakukan dengan menggunakan ambang batas, biasanya 0,5. Nilai probabilitas sebesar 0,5 terjadi ketika nilai z sama dengan nol. Artinya, batas keputusan ditentukan oleh persamaan $\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$, yang merupakan persamaan garis lurus pada ruang fitur dua dimensi (jam belajar dan persentase kehadiran). Oleh karena itu, meskipun probabilitas dibentuk melalui fungsi non-linier, batas pemisah antara mahasiswa lulus dan gagal tetap berbentuk linier.

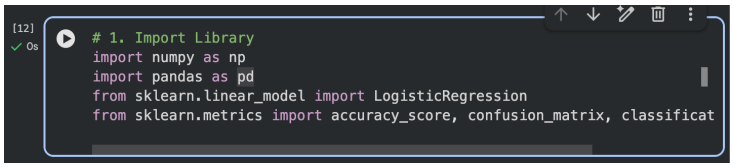
c. Lakukan inferensi dengan menunjukan tahapan solusinya

1. Study hours= 1.5, Attendance=58
2. Study hours= 5.5, Attendance=70

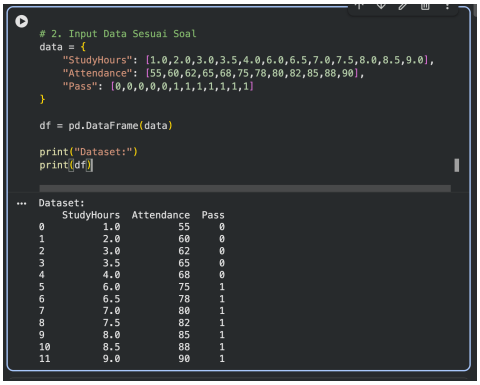
Jawab:

Untuk melakukan proses inferensi pada regresi logistik, berikut langkah-langkahnya dalam kode python:

1. Import library yang digunakan



2. Deklarasi variable yang akan digunakan sebagai dataset sesuai dengan soal



3. Memisahkan kolom fitur dan target

```python
# 3. Pisahkan Fitur dan Target
X = df[["StudyHours", "Attendance"]]
y = df["Pass"]
```

4. Membuat dan melatih model, lalu tampilkan nilai koefisien dan interceptnya

```python
# 4. Buat dan Latih Model
model = LogisticRegression()
model.fit(X, y)
```

```
LogisticRegression()
```

```python
# 5. Tampilkan Koefisien dan Intercept
print("Intercept (β0):", model.intercept_[0])
print("Koefisien (β1, β2):", model.coef_[0])
```

```
Intercept (β0): -49.068530110334095
Koefisien (β1, β2): [0.18358551 0.67356464]
```

5. Inferensi sesuai dengan kasus di atas

```python
# 6. Inferensi untuk Data Baru

# Kasus 1
data1 = np.array([[1.5, 58]])
prob1 = model.predict_proba(data1)[0][1]
pred1 = model.predict(data1)[0]

# Kasus 2
data2 = np.array([[5.5, 70]])
prob2 = model.predict_proba(data2)[0][1]
pred2 = model.predict(data2)[0]

print("--- Inferensi ---")
print("StudyHours=1.5, Attendance=58")
print("Probabilitas Lulus:", prob1)
print("Prediksi:", pred1)

print("\nStudyHours=5.5, Attendance=70")
print("Probabilitas Lulus:", prob2)
print("Prediksi:", pred2)
```

```
--- Inferensi ---
StudyHours=1.5, Attendance=58
Probabilitas Lulus: 5.968305472858739e-05
Prediksi: 0

StudyHours=5.5, Attendance=70
Probabilitas Lulus: 0.2871461672305763
Prediksi: 0
```

Berdasarkan hasil pemodelan regresi logistik, dilakukan pengujian terhadap dua kombinasi data baru untuk melihat probabilitas kelulusan mahasiswa.

Pada kasus pertama, dengan jam belajar 1,5 jam per hari dan tingkat kehadiran 58%, diperoleh probabilitas kelulusan sebesar $5{,}97 \times 10^{-5}$ atau sekitar 0,006%. Nilai ini sangat mendekati nol dan jauh di bawah ambang batas klasifikasi 0,5. Oleh karena itu, model memprediksi mahasiswa tersebut gagal (0). Hasil ini konsisten secara logis karena jam belajar dan tingkat kehadiran yang rendah memang cenderung berkorelasi dengan kegagalan.

Pada kasus kedua, dengan jam belajar 5,5 jam dan tingkat kehadiran 70%, probabilitas kelulusan yang diperoleh sebesar 0,287 atau sekitar 28,7%. Meskipun nilainya lebih tinggi dibandingkan kasus pertama, probabilitas ini masih berada di bawah threshold 0,5, sehingga model tetap memprediksi mahasiswa tersebut gagal (0). Hal ini menunjukkan bahwa berdasarkan pola data pelatihan, kombinasi 5,5 jam belajar dan 70% kehadiran belum cukup kuat untuk menggeser keputusan ke kategori lulus.

Secara keseluruhan, hasil ini menunjukkan bahwa model regresi logistik membentuk batas keputusan yang cukup tegas, di mana probabilitas kelulusan baru meningkat signifikan pada kombinasi jam belajar dan kehadiran yang lebih tinggi.

5. LO 2: a model problem using machine learning methods (20%)

Sebuah web streaming film hendak melakukan klasifikasi terhadap review film yang diterima. Anda diminta untuk membuat model Naive Bayes berdasarkan data berikut

ID	Review text	Sentiment
T1	movie was excellent	Positive
T2	excellent acting and story	Positive
T3	great movie with good acting	Positive
T4	amazing storyline and performances	Positive
T5	a movie with a nice scenery	Positive
T6	boring and slow story	Negative
T7	bad acting and bad CGI	Negative
T8	terrible movie experience	Negative
T9	movie was dissapointing	Negative
T10	bad ending ever	Negative

Lakukan inference terhadap test data berikut:

a. amazing experience

b. terrible acting

c. slow movie

d. enjoyable movie

Note: Gunakan kode python pada soal ini dan sertakan kode beserta penjelasan langkah-langkah hasil prediksi dalam format zip.

Jawab:

1. Import library yang digunakan dan siapkan dataset

```
import math
from collections import defaultdict

# Kode ini digunakan untuk menyiapkan dataset training yang akan dipakai
# dalam membangun model Naive Bayes. Setiap data terdiri dari teks review
# dan label sentimen (Positive atau Negative) sebagai kelas target.

# 1. Data Training
train_data = [
    ("movie was excellent", "Positive"),
    ("excellent acting and story", "Positive"),
    ("great movie with good acting", "Positive"),
    ("amazing storyline and performances", "Positive"),
    ("a movie with a nice scenery", "Positive"),
    ("boring and slow story", "Negative"),
    ("bad acting and bad CGI", "Negative"),
    ("terrible movie experience", "Negative"),
    ("movie was dissapointing", "Negative"),
    ("bad ending ever", "Negative"),
]

print(train_data)
```

```
[('movie was excellent', 'Positive'), ('excellent acting and story', 'Positive'), ('great movie with good acting', 'Positive')
```

2. Menyiapkan fungsi untuk mengubah teks menjadi lowercase dan dipecah menjadi per kata sebagai proses preprocessing

```
# Fungsi ini digunakan untuk melakukan preprocessing sederhana pada teks.
# Teks diubah menjadi huruf kecil (lowercase) agar konsisten,
# lalu dipecah menjadi daftar kata (token) menggunakan spasi sebagai pemisah.

# 2. Preprocessing
def tokenize(text):
    return text.lower().split()
```

3. Menyiapkan struktur data untuk melatih model Naive Bayes

```
# Bagian ini menyiapkan struktur data untuk melatih model Naive Bayes.
# vocab digunakan untuk menyimpan seluruh kata unik,
# word_counts menyimpan frekuensi kata per kelas,
# class_counts menghitung jumlah data per kelas,
# dan total_words menghitung total kata dalam setiap kelas.

# 3. Train Naive Bayes
vocab = set()
word_counts = {"Positive": defaultdict(int), "Negative": defaultdict(int)}
class_counts = {"Positive": 0, "Negative": 0}
total_words = {"Positive": 0, "Negative": 0}

print("word_counts", word_counts)
print("class_counts", class_counts)
print("total_words", total_words)
```

```
word_counts {'Positive': defaultdict(<class 'int'>, {}), 'Negative': defaultdict(<class 'int'>, {})}
class_counts {'Positive': 0, 'Negative': 0}
total_words {'Positive': 0, 'Negative': 0}
```

4. Menghitung frekuensi munculnya setiap kata

```
# Bagian ini menghitung frekuensi kemunculan setiap kata pada masing-masing kelas.
# Setiap review dipecah menjadi kata-kata, lalu dihitung jumlah kemunculannya
# untuk kelas Positive dan Negative. Selain itu, kita juga menyimpan jumlah
# total kata dan ukuran vocabulary (jumlah kata unik).

# Hitung frekuensi kata
for text, label in train_data:
    class_counts[label] += 1
    words = tokenize(text)
    for word in words:
        vocab.add(word)
        word_counts[label][word] += 1
        total_words[label] += 1

vocab_size = len(vocab)

print("vocab_size", vocab_size)
```

```
vocab_size 24
```

5. Menyiapkan fungsi untuk melakukan prediksi terhadap teks baru

```
# Fungsi ini digunakan untuk melakukan prediksi sentimen terhadap teks baru.
# Model menghitung skor untuk setiap kelas berdasarkan prior probability
# dan probabilitas kemunculan kata (dengan Laplace smoothing), lalu memilih
# kelas dengan skor tertinggi sebagai hasil prediksi.

# 4. Prediction Function
def predict(text):
    words = tokenize(text)

    scores = {}

    for label in ["Positive", "Negative"]:
        # prior probability
        score = math.log(class_counts[label] / len(train_data))

        for word in words:
            # Laplace smoothing
            word_freq = word_counts[label][word] + 1
            score += math.log(word_freq / (total_words[label] + vocab_size))

        scores[label] = score

    return max(scores, key=scores.get)
```

6. Pengujian model dengan data baru

```
# Bagian ini digunakan untuk menguji model dengan data baru yang belum pernah
# dilihat sebelumnya. Setiap kalimat test akan diprediksi sentimennya menggunakan
# fungsi predict(), lalu hasil klasifikasinya ditampilkan.

# 5. Test Data
test_data = [
    "amazing experience",
    "terrible acting",
    "slow movie",
    "enjoyable movie"
]

for test in test_data:
    print(f"{test} --> {predict(test)}")
```

```
amazing experience --> Negative
terrible acting --> Negative
slow movie --> Negative
enjoyable movie --> Positive
```

Berdasarkan hasil pengujian, tiga dari empat kalimat uji diklasifikasikan sebagai Negative, yaitu “amazing experience”, “terrible acting”, dan “slow movie”. Prediksi negatif pada “terrible acting” dan “slow movie” dapat dijelaskan karena kata terrible dan slow muncul dalam data latih berlabel negatif, sehingga memberikan kontribusi probabilitas lebih besar pada kelas negatif. Sementara itu, “amazing experience” diprediksi negatif kemungkinan karena kata experience lebih sering muncul pada data negatif dan pengaruhnya lebih dominan dibanding kata amazing, meskipun secara umum kata tersebut bernuansa positif.

Sebaliknya, “enjoyable movie” diprediksi sebagai Positive. Walaupun kata enjoyable tidak muncul dalam data latih, Laplace smoothing memungkinkan kata tersebut tetap memiliki probabilitas kecil pada kedua kelas. Selain itu, kata movie lebih sering muncul dalam data positif, sehingga secara keseluruhan skor probabilitas kelas positif menjadi lebih tinggi dibandingkan negatif. Hal ini menunjukkan bahwa model Naive Bayes sangat bergantung pada distribusi kata dalam data latih dan dapat menghasilkan prediksi yang berbeda dari intuisi manusia jika konteks kata terbatas. Kode lengkapnya dapat diakses melalui link berikut https://drive.google.com/file/d/1CXIvIMjWbKLsf4wlic7sQORIVLGA1ZMJ/view?usp=sharing.

Referensi:

1. *https://www.sciencedirect.com/topics/mathematics/curse-of-dimensionality*
2. *https://machinelearningmastery.com/how-to-handle-big-p-little-n-p-n-in-machine-learning/*
3. *Lecture Notes Week 6*
4. *Lecture Notes Week 8*
5. *Lecture Notes Week 4*
6. *Lecture Notes Week 9*