Supervised Learning - Classification

by Thio Perdana

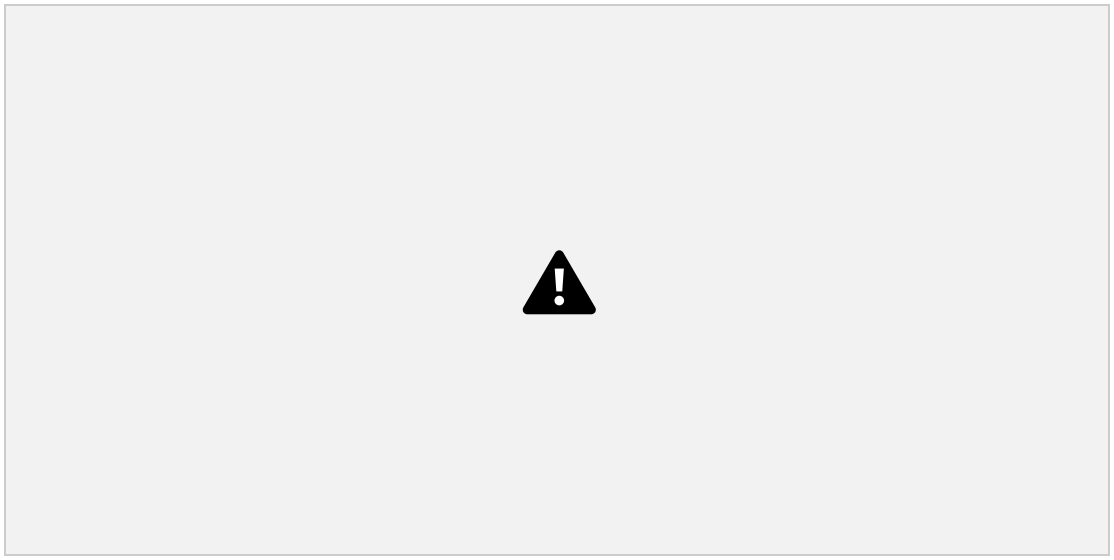
OUTLINE

Tipe-Tipe Klasifikasi

- **Logistic Regression**
- **Naive Bayes**
- **KNN**
- **Decision Tree Classification**
- **Support Vector Machine**

Klasifikasi dengan Python

Classification atau Klasifikasi adalah proses untuk mengenali, mengerti, dan mengumpulkan obyek atau idea yang memiliki kesamaan ke dalam suatu kategori. Algoritma klasifikasi dalam machine learning memanfaatkan masukan dari data training dengan tujuan untuk memperkirakan kebolehjadian atau probabilitas suatu data masuk kedalam suatu kategori. Salah satu contoh paling sederhana bagaimana algoritma klasifikasi digunakan untuk memisahkan email yang termasuk pada spam atau tidak.



Secara sederhana klasifikasi digunakan untuk mengenali pola. Data yang memiliki kesamaan pola akan dikumpulkan dalam suatu grup atau kateogori. Pada supervised learning, hal ini dilakukan dengan menggunakan data label sebagai acuan. Dengan melihat pola pada fitur-fitur yang dimiliki oleh sebuah instance data baru maka model akan berusaha mencari kemiripan pola dengan data latih lalu setelah itu memberikan label.

Jenis Klasifikasi

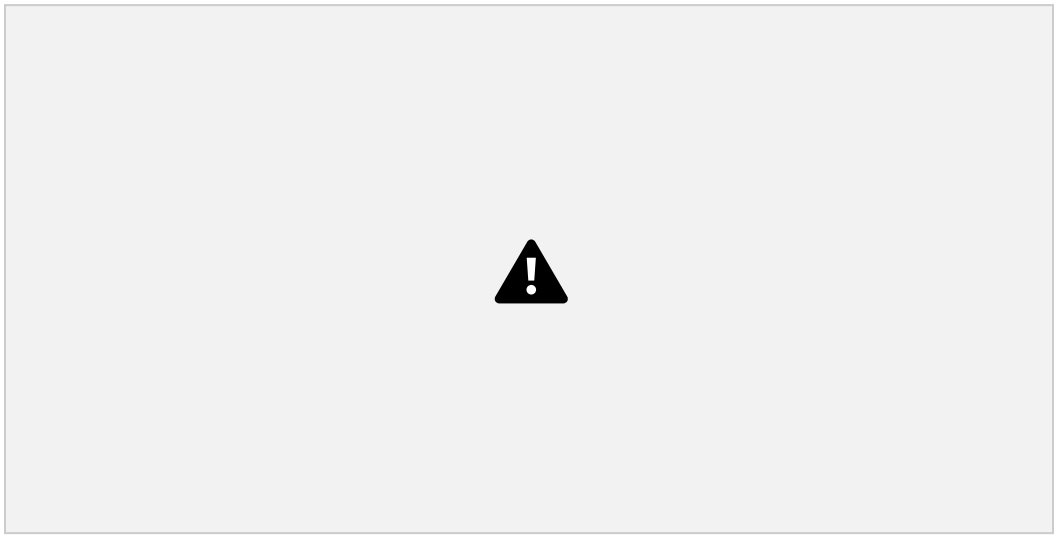
Terdapat 5 buah algoritma klasifikasi yang akan kita bahas pada kesempatan kali ini, yaitu:

1. Logistik Regression
2. Naive Bayes
3. K-Nearest Neighbors
4. Decision Tree

5. Support Vector Machine

Logistic Regression

Memiliki nama regresi tidak membuat logistic regression menjadi algoritma regresi. Logistic Regression adalah algoritma klasifikasi yang memanfaatkan logistic function untuk mengkategorikan data. Logistic function atau logistic function atau sering disebut dengan nama fungsi sigmoid adalah fungsi yang hasil keluarannya adalah 1 atau 0. Karena itu pada algoritma ini label yang bisa disematkan hanyalah 2, besar - kecil, tinggi - rendah, mahal - murah, atau membeli - tidak membeli.



Persamaan sigmoid berbentuk

$$\varnothing(x) = \frac{1}{1 + e^{-x}}$$

Logisitic Regression dapat bekerja dengan feature lebih dari satu, berbeda dengan saudaranya--linear regression. Akan tetapi kekurangannya adalah dataset yang kita miliki haruslah memiliki label yang binary seperti yang disebutkan di atas. Contoh penerapan algoritma ini adalah saat menangani kasus **Prediksi Kelulusan:** Menggunakan Regresi Logistik untuk memprediksi kemungkinan kelulusan siswa berdasarkan variabel seperti jumlah jam belajar dan nilai ujian.

Bagaimana Regresi Logistik Bekerja

Probabilitas dan Keputusan: Regresi Logistik menghitung probabilitas bahwa suatu instance data masuk ke dalam kelas tertentu. Jika probabilitas lebih besar dari ambang tertentu (biasanya 0,5), instance tersebut diklasifikasikan ke dalam kelas tersebut.

Pelatihan Model: Model dilatih dengan mengoptimalkan parameter (koefisien) agar memberikan hasil probabilitas yang sesuai dengan kelas yang sebenarnya.

Kelebihan dan Kekurangan Regresi Logistik:

Kelebihan:

Mudah diinterpretasi.

Cocok untuk klasifikasi biner.

Kinerja baik untuk data yang linier terpisah.

Kekurangan:

Tidak bekerja baik untuk data yang tidak linier terpisah.

Rentan terhadap overfitting.

Mengasumsikan hubungan linier antara fitur dan log-odds.

Naive Bayes

Naive Bayes adalah metode yang menggunakan prinsip probability untuk membuat model prediksi klasifikasi. Dengan memanfaatkan data tentang kejadian masa lalu, model bisa membuat perkiraan yang akan terjadi di masa depan. Metode ini menghitung probability suatu kejadian, dan bisa berubah bila ada informasi pendukung tambahan yang disediakan.

Keunggulan NB adalah sifatnya yang efektif dan cepat untuk mengolah data berjumlah besar. Karena kelebihan itulah NB salah satu algoritma yang sering dipakai untuk melakukan klasifikasi data text, misal terkait sentiment analysis maupun email spamming. Hanya saja kekurangan terbesar dari algoritma ini seperti pada namanya, Naive. Algoritma ini memperlakukan semua fitur sama. melihat hanya dari

probabilitasnya. Jadi untuk data yang lumayan komplek Naive Bayes seringkali gagal dalam melakukan klasifikasi yang baik.

Persamaan probabilitas kondisional yang menjadi dasar algoritma ini dapat dilihat di bawah ini

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B) adalah **posterior.** Ini adalah nilai yang kita cari. Misalkan probabilitas seseorang terkena maag jika memakan cabai.

P(B|A) adalah **likelihood.** Ini adalah probabilitas untuk menemukan data baru melihat hipotesis awal kita. Misalkan berapa banyak pemakan cabai yang terkena maag

P(A) adalah **prior.** Ini adalah probabilitas hipotesis kita tanpa adanya tambahan informasi sebelumnya. Misalnya probabilitas orang terkena maag

P(B) adalah **marginal likelihood.** Ini adalah total probabilitas menemukan yang kita cari. Misalnya probabilitas orang suka makan cabai.

Bagaimana Naive Bayes Bekerja

Asumsi Kemandirian Fitur (Naivitas): Meskipun namanya "naive" (naif), algoritma ini bekerja dengan mengasumsikan bahwa setiap fitur dalam data tidak terkait satu sama lainnya. Meskipun ini terlalu sederhana untuk banyak kasus, namun dalam banyak situasi, asumsi ini cukup baik dan mempermudah perhitungan.

Klasifikasi dengan Probabilitas: Naive Bayes mengklasifikasikan data berdasarkan probabilitas. Misalnya, dalam klasifikasi spam atau non-spam email, algoritma menghitung probabilitas bahwa suatu email adalah spam atau non-spam berdasarkan kata-kata yang ada di dalamnya.

Jenis Naive Bayes

Multinomial Naive Bayes: Cocok untuk data kategori yang dihitung, seperti jumlah kata dalam suatu dokumen.

Gaussian Naive Bayes: Cocok untuk data numerik yang terdistribusi normal.

Bernoulli Naive Bayes: Cocok untuk data biner, seperti keberadaan atau ketidakberadaan suatu fitur.

Kelebihan dan Kekurangan Naive Bayes

Kelebihan:

Cepat dan efisien.

Baik untuk dataset besar.

Menangani fitur yang tidak relevan dengan baik.

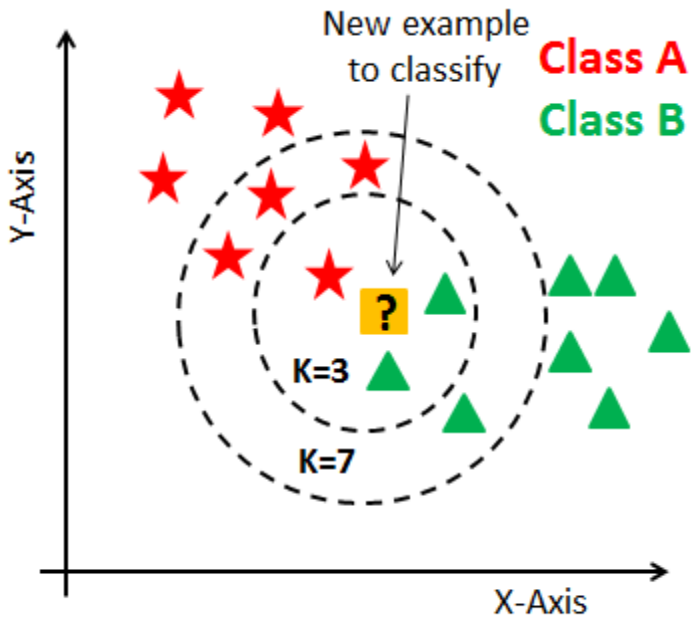
Kekurangan:

Asumsi independensi yang terlalu sederhana.

Kinerja dapat menurun jika ada fitur yang sangat terkait.

K-Nearest Neighbors

K-Nearest Neighbors adalah logaritma yang melakukan klasifikasi dengan melihat K tetangga yang berada di sekitar data, lalu klasifikasi data akan mengikuti klasifikasi tetangga terbanyak. Perumpamaannya seperti ini, misalkan kita baru saja membangun daerah di sebuah perbatasan RW. Untuk mengetahui kita masuk ke RW 1 atau RW dua kita menentukan kita akan melihat 5 rumah tetangga terdekat dengan kita (5 adalah nilai K nya). Setelah kita menentukan jumlahnya, kita cek untuk kelima tetangga tersebut masuk ke RW berapa. Ternyata 3 rumah RW 1 dan 2 rumah RW 2, maka kesimpulan yang bisa diambil adalah kita masuk ke RW 1.



Kelebihan dari algoritma ini adalah dia cukup fleksibel dan mudah dilakukan. Pengolahan untuk dataset yang besar juga relatif lebih dapat dilakukan. Akan tetapi, karena pentuan K itu penting maka beban komputasi jadi lebih tinggi dari algoritma lainnya.

Bagaimana K-NN Bekerja

Menentukan Parameter K: Pilih jumlah tetangga terdekat (K) yang akan digunakan dalam pengambilan keputusan.

Mengukur Jarak: Hitung jarak antara data yang akan diprediksi dengan semua data latih menggunakan metrik seperti Euclidean atau Manhattan.

Menentukan Tetangga Terdekat: Pilih K tetangga terdekat berdasarkan jarak yang diukur.

Voting untuk Kelas Mayoritas: Tentukan kelas mayoritas dari K tetangga tersebut, dan inilah kelas prediksi data.

Kelebihan dan Kekurangan K-NN

Kelebihan:

Sederhana dan mudah diimplementasikan.

Efektif untuk data dengan pola yang kompleks.

Kekurangan:

Rentan terhadap outlier.

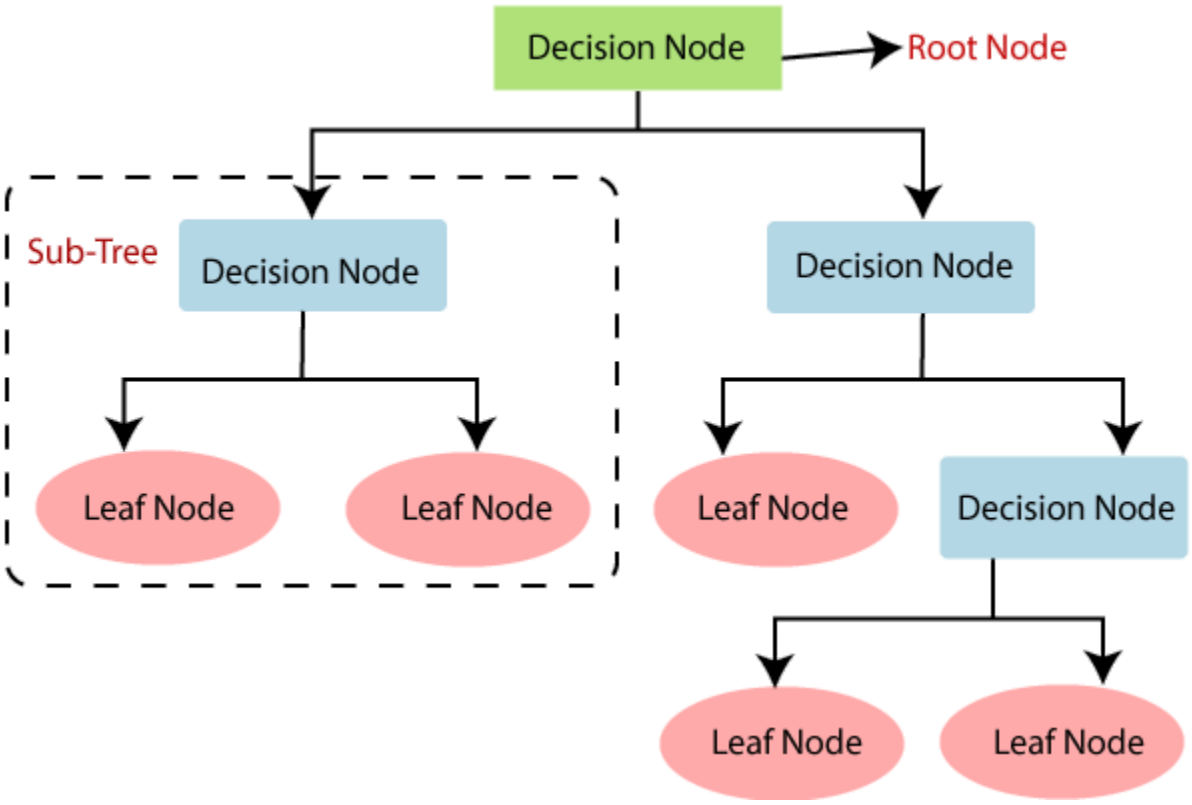
Kinerja menurun dengan dimensi fitur yang tinggi.

Pemilihan parameter K yang tepat krusial.

Decision Tree

Decision Tree pada klasifikasi punya dasar yang sama dengan pada regresi. Menggunakan struktur pohon untuk melakukan klasifikasinya. Decision Tree terdiri dari node (simpul), cabang, dan daun. Setiap

node mewakili keputusan atau tes pada fitur tertentu, cabang adalah hasil dari tes tersebut, dan daun menyatakan kelas atau nilai hasil.



Bagaimana Decision Tree Bekerja

Pemilihan Fitur: Decision Tree memilih fitur terbaik untuk membagi data berdasarkan kriteria seperti Gini Index atau Entropy. Fitur yang memberikan pemisahan yang paling baik dipilih.

Pembagian Data: Data dibagi ke dalam subset berdasarkan nilai fitur yang dipilih, dan proses ini diulang untuk setiap subset.

Membangun Struktur Pohon: Proses pemilihan fitur dan pembagian data diulang hingga mencapai kondisi berhenti, seperti kedalaman maksimum atau ukuran subset yang cukup kecil.

Langkah-langkah Implementasi

Pemahaman Data: Memahami data dan menentukan fitur serta variabel dependen (kelas) yang akan diprediksi.

Pemilihan Kriteria Pemisahan: Memilih kriteria (misalnya, Gini Index) untuk menentukan fitur terbaik pada setiap langkah.

Rekursif Membangun Pohon: Memulai dari root, secara rekursif membagi data dan membangun pohon hingga mencapai kondisi berhenti.

Pruning (Pemangkasan): Beberapa pohon mungkin menjadi terlalu kompleks dan rentan terhadap overfitting. Proses pemangkasan dapat diterapkan untuk mengurangi kompleksitas dan meningkatkan generalisasi.

Prediksi dan Evaluasi Model: Setelah pohon dibangun, model dapat digunakan untuk memprediksi kelas baru, dan kemudian dievaluasi menggunakan metrik seperti akurasi, presisi, dan recall.

Kelebihan dan Kekurangan Decision Tree

Kelebihan:

Mudah diinterpretasi dan visualisasi.

Cocok untuk data yang kompleks dan non-linier.

Tidak memerlukan normalisasi data.

Kekurangan:

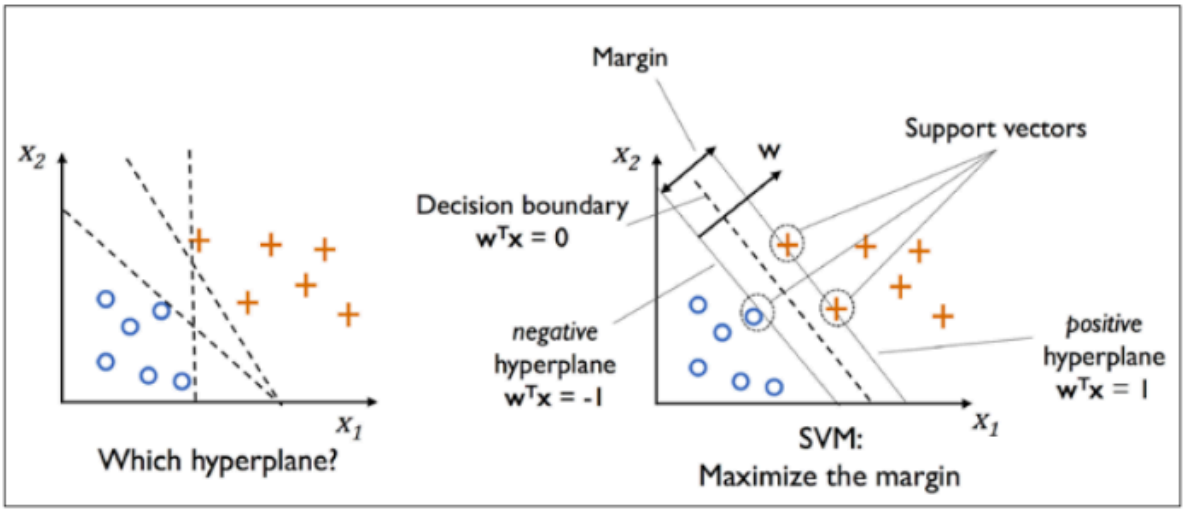
Rentan terhadap overfitting.

Tidak efektif untuk data dengan hubungan linier yang kompleks.

Sensitif terhadap perubahan kecil dalam data.

Support Vector Machine

Secara umum, konsep SVM hampir sama dengan SVR bedanya pada penggunaan hyperplane parameter. Pada SVM, hyperplane digunakan untuk memisahkan data. Margin yang paling besar akan dipilih untuk dijadikan pembatas saat melakukan prediksi kedepannya.



Kelebihannya dari algoritma ini adalah lebih efisien terkait memory dan mendukung dengan baik untuk data yang lebih dari 2 dimensi (fitur). Akan tetapi, sulit mendapatkan data probabilitas dari algoritma ini. Karena SVM murni menggunakan geometri untuk menghitung hyperplane terbaik. Karena itu juga, normalisasi data pada algoritma harus dilihat dengan lebih serius.

Bagaimana SVM Bekerja

Hyperplane: SVM mencari hyperplane (bidang pembatas) yang paling baik memisahkan dua kelas.

Margin: Margin adalah jarak antara hyperplane dan sampel terdekat dari masing-masing kelas. SVM berusaha memaksimalkan margin.

Support Vectors: Sampel yang berada tepat di batas margin disebut vektor pendukung (support vectors).

Kernel Trick: SVM dapat menangani data yang tidak linear dengan menggunakan fungsi kernel untuk mentransformasikan data ke dimensi yang lebih tinggi.

Kelebihan dan Kekurangan SVM

Kelebihan:

Efektif dalam ruang fitur berdimensi tinggi.

Bekerja baik dengan data yang memiliki batas keputusan yang kompleks.

Kekurangan:

Memerlukan perhatian khusus pada pemilihan kernel.

Tidak efisien untuk set data yang sangat besar.

Klasifikasi dengan Python

Pada kesempatan kali ini kita akan mengolah data pasien diabetes di India. Untuk tutorial kali ini silahkan unduh dataset di bawah ini

unduh dataset

Kita akan membuat model logistik regression

pertama kita buka dan tampilkan dataset kita

```python
import pandas as pd

df = pd.read_csv("diabetes.csv")
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Karena jumlah data yang banyak, mari kita coba untuk melakukan pengecekan terhadap nilai nan

```python
df.isna().sum()
```

✓ 0.7s

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Data sudah bersih dari nilai NaN.

Karena pada logistic regression kita dapat memilih lebih dari satu fitur, maka pada tutorial kali ini kita akan mengambil kolom *pregnancies* hingga *age* untuk mengisi sumbu x dan kolom outcome untuk mengisi sumbu y

```
X = df.iloc[:,:8]
y = df.iloc[:,8:9]
```

Karena data telah disiapkan, maka yang perlu kita lakukan sekarang adalah membuat model kita.

Gunakan model untuk logistic regression.

Sebelum itu kita akan bagi data kita menjadi data training : test dengan perbandingan 8:2

```
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=16)
```

```
from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
lg = LogisticRegression()

# fit the model with data
lg.fit(X_train,y_train)

#
y_pred=lg.predict(X_test)
```

Model sudah selesai kita buat, Apakah model ini model yang bagus?

Kita masih belum bisa menjawabnya, untuk itu kita memerlukan evaluasi. Tapi berbeda dengan Linear regression, kita tidak akan menggunakan R2, akan tetapi melihat nilai AUC.

AUC singkatan dari *Area Under Curve,* sebuah daerah dalam kuva ROC. Semakin mendekati nilai 1 untuk AUC, maka semakin baik juga model dalam menggambarkan dataset yang kita miliki.

Hanya saja yang harus diingat, AUC nilai 1 bisa berarti model bagus atau data kita overfitting.

```
from sklearn import metrics


auc = metrics.accuracy_score(y_test, y_pred)

print(f'Nilai AUC : {auc}')
```

Hasilnya adalah:

Nilai AUC : 0.8181818181818182

Itu merupakan nilai yang cukup bagus. setelah model selesai kita bisa menggunakan model tersebut untuk mengklasifikasin jika ada data baru yang masuk

---