Dalam KNN, lebih baik mana melakukan normalisasi dulu atau feature selection sebelum melatih model?

Dalam algoritma K-Nearest Neighbors (KNN), **normalisasi** umumnya lebih penting dilakukan terlebih dahulu sebelum **feature selection**, dan ini tergantung pada alasan di balik masing-masing langkah. Berikut adalah alasan mengapa normalisasi lebih baik dilakukan terlebih dahulu:

**1. Pentingnya Normalisasi pada KNN:**

* **Skala Fitur yang Berbeda:** KNN adalah algoritma berbasis jarak (distance-based algorithm). Jika fitur memiliki skala yang sangat berbeda (misalnya, satu fitur berkisar antara 0 hingga 1, sementara yang lain berkisar antara 1000 hingga 10000), fitur dengan skala yang lebih besar akan mendominasi perhitungan jarak, yang bisa mempengaruhi hasil prediksi secara signifikan.
* **Normalisasi Membantu Membuat Perbandingan yang Adil:** Dengan melakukan normalisasi (misalnya, menggunakan Min-Max Scaling atau Standardization), semua fitur akan berada dalam skala yang setara, yang memungkinkan KNN untuk memberikan perhatian yang lebih adil terhadap setiap fitur dalam pengukuran jarak.

**2. Feature Selection Setelah Normalisasi:**

* **Meminimalkan Fitur yang Tidak Relevan:** Setelah normalisasi, Anda akan lebih mudah mengevaluasi fitur-fitur yang relevan dan tidak relevan. Misalnya, Anda dapat menggunakan teknik seperti **mutual information**, **correlation analysis**, atau **p-value testing** untuk memilih fitur yang paling signifikan tanpa khawatir bahwa perbedaan skala fitur akan mempengaruhi hasil.
* **Mengurangi Dimensi:** Feature selection setelah normalisasi dapat mengurangi dimensi data tanpa kehilangan informasi yang penting, sehingga meningkatkan kinerja model dan mengurangi waktu training.

**Kesimpulannya:**

**Lakukan normalisasi terlebih dahulu**, baru kemudian lakukan feature selection. Normalisasi akan memastikan bahwa fitur memiliki kontribusi yang seimbang dalam perhitungan jarak, sementara feature selection dapat membantu mengidentifikasi fitur yang paling penting setelah semua fitur berada pada skala yang setara.

Oke baik, berarti lakukan normalisasi dulu baru feature selection ya. sekarang pertanyaan saya. Split data (membagi dataset menjadi dataset training dan testing) itu dilakukan sebelum normalisasi atau sesudah normalisasi?

Untuk pertanyaan ini, **data split** (membagi dataset menjadi data training dan testing) sebaiknya dilakukan **sebelum normalisasi**. Berikut penjelasannya:

**Alasan Melakukan Split Sebelum Normalisasi:**

1. **Mencegah Kebocoran Data (Data Leakage):**
   * Jika Anda melakukan normalisasi sebelum membagi data, statistik seperti mean dan standard deviation (untuk metode seperti standardization) akan dihitung berdasarkan seluruh dataset, termasuk data uji (testing data). Ini akan menyebabkan kebocoran informasi dari data testing ke dalam data training, yang berpotensi mengurangi generalisasi model dan menambah bias.
   * Dengan membagi data terlebih dahulu, normalisasi dilakukan hanya pada data training, dan statistik normalisasi (misalnya, mean dan standard deviation) dihitung hanya berdasarkan data training. Kemudian, statistik ini diterapkan pada data testing untuk menjaga integritas evaluasi model.
2. **Menjaga Keaslian Distribusi Data:**
   * Proses normalisasi yang dilakukan setelah data dibagi memastikan bahwa data uji tetap mencerminkan distribusi yang realistis sesuai dengan data training, yang lebih mencerminkan bagaimana model akan bekerja pada data yang tidak terlihat sebelumnya.

Berarti alurnya split data dulu lalu dilakukan normalisasi, selanjutnya dilakukan feature selection, baru setelah itu melatih model?

Ya, alur yang benar adalah seperti berikut:

1. **Split Data**: Pertama, bagi dataset Anda menjadi data training (training) dan data testing (testing).
2. **Normalisasi**: Lakukan normalisasi pada data training (training set). Anda dapat menggunakan teknik seperti Min-Max Scaling atau Standardization. Setelah itu, terapkan transformasi yang sama (menggunakan parameter yang dihitung dari data training) pada data testing (testing set).
3. **Feature Selection**: Setelah normalisasi, lakukan feature selection pada data training. Anda dapat menggunakan metode seperti **mutual information**, **PCA**, atau **correlation analysis** untuk memilih fitur yang paling relevan. Pastikan proses feature selection hanya dilakukan menggunakan data training, karena Anda tidak ingin mengungkapkan informasi dari data testing kepada model.
4. **Melatih Model**: Setelah feature selection, latih model Anda menggunakan data training yang sudah dinormalisasi dan dipilih fitur-fitur relevannya.
5. **Evaluasi Model**: Setelah model dilatih, evaluasi performanya menggunakan data testing yang sudah dinormalisasi dan dipilih fitur-fitur relevannya.

1. Normalisasi itu dilakukan pada dataset training saja atau dataset testing juga?

2. Feature Selection itu dilakukan pada dataset training saja atau dataset testing juga?

Berikut adalah jawaban untuk kedua pertanyaan Anda:

**1. Normalisasi:**

**Normalisasi hanya dilakukan pada dataset training,** kemudian diterapkan pada dataset testing.

* **Penjelasan:** Normalisasi atau standariasi pada dasarnya melibatkan perhitungan statistik (seperti mean, standard deviation, atau min-max scaling) berdasarkan dataset training. Ini berarti, Anda menggunakan informasi dari data training saja untuk mengubah skala data. Setelah itu, transformasi yang sama (misalnya, menggunakan mean dan standard deviation yang dihitung dari data training) diterapkan pada data testing.
* **Tujuan:** Hal ini mencegah data testing dari "terbocornya" informasi yang hanya seharusnya ada di data training, yang bisa menyebabkan bias pada evaluasi model dan memperburuk generalisasi model pada data nyata.

**2. Feature Selection:**

**Feature selection hanya dilakukan pada dataset training,** dan tidak dilakukan pada dataset testing.

* **Penjelasan:** Feature selection bertujuan untuk memilih fitur yang paling relevan untuk model, dan seharusnya hanya menggunakan data yang tersedia di dataset training. Jika feature selection dilakukan dengan melibatkan dataset testing, itu akan menyebabkan kebocoran informasi, karena fitur yang dipilih bisa dipengaruhi oleh data testing yang seharusnya tidak digunakan untuk melatih model.
* **Tujuan:** Agar model hanya belajar dari data training tanpa melibatkan data testing yang seharusnya menjadi data yang tidak terlihat (unseen), sehingga hasil evaluasi lebih akurat.

**Ringkasan Alurnya:**

1. **Split Data**: Bagi dataset menjadi data training dan testing.
2. **Normalisasi**: Lakukan normalisasi hanya pada data training, kemudian terapkan transformasi pada data testing.
3. **Feature Selection**: Lakukan feature selection hanya pada data training.
4. **Latih Model**: Latih model menggunakan data training yang sudah dinormalisasi dan difilter berdasarkan feature selection.
5. **Evaluasi Model**: Evaluasi model menggunakan data testing yang sudah dinormalisasi.

Dengan langkah-langkah ini, Anda dapat memastikan model yang lebih akurat dan dapat digeneralisasi dengan baik tanpa kebocoran data.

Berarti?

1. Normalisasi dilakukan pada kedua dataset (training dan testing)

2. Feature Selection hanya dilakukan pada dataset training

Sebenarnya, ada sedikit klarifikasi yang perlu dilakukan pada pernyataan Anda:

**1. Normalisasi:**

* **Normalisasi dilakukan hanya pada dataset training**, namun transformasi yang diperoleh dari dataset training (misalnya, mean, standard deviation, atau nilai min-max) **dapat diterapkan pada dataset testing**.
* **Penjelasan:**
  + Pada dataset training, Anda menghitung parameter normalisasi (misalnya, mean dan standard deviation jika menggunakan standardization).
  + Setelah itu, transformasi normalisasi yang sama diterapkan pada dataset testing menggunakan parameter yang dihitung dari dataset training.
* Jadi, **normalisasi dilakukan pada keduanya**, tetapi statistik yang digunakan untuk normalisasi hanya dihitung dari data training, bukan data testing.

**2. Feature Selection:**

* **Feature selection hanya dilakukan pada dataset training**.
* **Penjelasan:**
  + Feature selection bertujuan untuk memilih fitur terbaik untuk model dan harus dilakukan hanya pada data training.
  + Setelah fitur yang relevan dipilih dari data training, model akan dilatih menggunakan fitur-fitur tersebut.
  + Data testing hanya digunakan untuk evaluasi model, tidak untuk pemilihan fitur.

**Ringkasan:**

1. **Normalisasi:**
   * Dilakukan pada dataset training dan transformasi diterapkan pada dataset testing menggunakan parameter yang dihitung dari dataset training.
2. **Feature Selection:**
   * Dilakukan hanya pada dataset training untuk memilih fitur yang paling relevan.

Dengan cara ini, Anda menjaga integritas data testing, yang seharusnya hanya digunakan untuk evaluasi model dan tidak memengaruhi tahap training.

Saya masih bingung. Jika normalisasi dilakukan hanya pada dataset training, namun transformasi yang diperoleh dari dataset training (misalnya, mean, standard deviation, atau nilai min-max) dapat diterapkan pada dataset testing.

Bukannya sama aja dengan normalisasi dilakukan pada kedua dataset (training dan testing)? Atau pemahaman saya yang salah?

**Perbedaan antara Normalisasi pada Training dan Testing**

**Normalisasi** dalam konteks machine learning sering melibatkan transformasi data, misalnya dengan menggunakan **min-max scaling** atau **standardization (z-score)**. Tujuannya adalah untuk mengubah fitur-fitur ke dalam skala yang lebih konsisten, karena algoritma seperti KNN sangat tergantung pada perhitungan jarak antar titik data.

Namun, ada perbedaan antara **menghitung statistik** untuk normalisasi dan **menerapkan transformasi** pada data. Inilah yang membedakan proses pada data training dan testing:

1. **Pada dataset training**:
   * Anda **menghitung statistik** yang dibutuhkan untuk normalisasi, seperti mean dan standard deviation (untuk standar deviasi) atau nilai min dan max (untuk min-max scaling).
   * Kemudian Anda **melakukan normalisasi** pada dataset training menggunakan statistik yang dihitung tersebut.
2. **Pada dataset testing**:
   * **Tidak ada perhitungan ulang statistik** berdasarkan data testing. Statistik yang dihitung dari data training (misalnya mean dan standard deviation) digunakan untuk **menerapkan transformasi yang sama** pada data testing.
   * **Transformasi yang sama diterapkan pada data testing**, tetapi **statistik untuk normalisasi** hanya berasal dari data training, bukan data testing.

**Kenapa Ini Penting?**

Jika kita melakukan normalisasi pada **kedua dataset (training dan testing)** dengan cara yang terpisah (yaitu, menghitung mean dan standard deviation dari masing-masing set secara independen), kita bisa menghadapi masalah **data leakage** atau **perbedaan distribusi** antara training dan testing. Ini akan membuat model tidak realistis dalam generalisasinya.

**Ringkasan:**

* **Normalisasi pada training**: Menghitung statistik (mean, standard deviation, atau min-max) berdasarkan data training dan melakukan normalisasi pada data training.
* **Normalisasi pada testing**: Tidak menghitung statistik baru, tetapi hanya **menerapkan** transformasi yang sudah dihitung pada data training ke data testing.

Dengan kata lain, **normalisasi tetap dilakukan pada kedua dataset**, tetapi statistik yang digunakan untuk normalisasi berasal dari **data training** dan hanya diterapkan ke **data testing** tanpa perhitungan ulang.

Saya akan menggunakan normalisasi dengan min-max scaling.

Berikan saya contoh kode apa bedanya:

Cell 1: Normalisasi pada training: Menghitung statistik (mean, standard deviation, atau min-max) berdasarkan data training dan melakukan normalisasi pada data training. Normalisasi pada testing: Tidak menghitung statistik baru, tetapi hanya menerapkan transformasi yang sudah dihitung pada data training ke data testing.

Cell 2: Melakukan normalisasi pada kedua dataset

Cell 1: Normalisasi pada Data Training dan Testing (Statistik dari Data Training)

from sklearn.preprocessing import MinMaxScaler

import numpy as np

# Contoh data (misal, 2 fitur, 5 data)

data\_training = np.array([[1, 2], [2, 3], [3, 4], [4, 5], [5, 6]])

data\_testing = np.array([[6, 7], [7, 8], [8, 9]])

# Membuat objek MinMaxScaler

scaler = MinMaxScaler()

# Melakukan fitting dan transformasi pada data training

data\_training\_normalized = scaler.fit\_transform(data\_training)

# Menerapkan transformasi yang sama pada data testing

data\_testing\_normalized = scaler.transform(data\_testing)

# Menampilkan hasil normalisasi

print("Data Training yang telah dinormalisasi:")

print(data\_training\_normalized)

print("\nData Testing yang telah dinormalisasi:")

print(data\_testing\_normalized)

**Penjelasan:**

* Pada kode ini, **fit\_transform()** digunakan untuk menghitung statistik (min dan max) berdasarkan **data training**, lalu melakukan normalisasi.
* **transform()** diterapkan pada **data testing** menggunakan parameter yang sama dari **data training** tanpa menghitung ulang statistik dari data testing.

Cell 2: Normalisasi pada Kedua Dataset (Training dan Testing)

from sklearn.preprocessing import MinMaxScaler

import numpy as np

# Contoh data (misal, 2 fitur, 5 data)

data\_training = np.array([[1, 2], [2, 3], [3, 4], [4, 5], [5, 6]])

data\_testing = np.array([[6, 7], [7, 8], [8, 9]])

# Membuat objek MinMaxScaler

scaler = MinMaxScaler()

# Melakukan fitting dan transformasi pada data training dan testing

data\_training\_normalized = scaler.fit\_transform(data\_training)

data\_testing\_normalized = scaler.fit\_transform(data\_testing) # Perhitungan statistik baru untuk data testing

# Menampilkan hasil normalisasi

print("Data Training yang telah dinormalisasi:")

print(data\_training\_normalized)

print("\nData Testing yang telah dinormalisasi:")

print(data\_testing\_normalized)

**Penjelasan:**

* Pada kode ini, **fit\_transform()** diterapkan pada **data training** untuk menghitung statistiknya dan melakukan normalisasi.
* Namun, pada **data testing**, **fit\_transform()** digunakan juga, yang berarti **statistik baru** dihitung berdasarkan **data testing**, yang bisa menyebabkan kebocoran informasi dan perbedaan distribusi yang tidak diinginkan antara training dan testing.

**Perbedaan Utama:**

* **Cell 1**: **Statistik hanya dihitung dari data training**, dan transformasi yang sama diterapkan pada data testing. Ini adalah cara yang benar untuk menjaga agar data testing tetap "tidak terlihat" selama training.
* **Cell 2**: **Statistik dihitung ulang pada data testing**, yang mengarah pada kebocoran informasi dan hasil yang kurang representatif untuk model yang sebenarnya.

Bagaimana saya membuat promt normalisasinya untuk cell 1 dan 2

Cell 1

"Implement the Min-Max scaling normalization on a training dataset and a testing dataset. First, fit the MinMaxScaler using the training dataset to compute the min and max values. Then, apply the same transformation (scaling) to the training dataset using the fitted scaler. Afterward, apply the same transformation to the testing dataset using the parameters derived from the training dataset, ensuring no recalculation of min and max for the testing data."

"Lakukan normalisasi dengan Min-Max scaling pada dataset training dan dataset testing. Pertama, fit MinMaxScaler menggunakan dataset training untuk menghitung nilai min dan max. Kemudian, terapkan transformasi (skala) yang sama pada dataset training menggunakan scaler yang telah dihitung. Setelah itu, terapkan transformasi yang sama pada dataset testing menggunakan parameter yang dihitung dari dataset training, pastikan tidak menghitung ulang nilai min dan max untuk data testing."

Cell 2

"Perform Min-Max scaling normalization on both the training dataset and the testing dataset. Apply the MinMaxScaler to both datasets separately, meaning compute the min and max values for each dataset individually using fit\_transform(). This will allow each dataset to be scaled independently, which is not the correct approach for preserving model integrity and generalization."

"Lakukan normalisasi dengan Min-Max scaling pada kedua dataset, yaitu dataset training dan dataset testing. Terapkan MinMaxScaler pada kedua dataset secara terpisah, yang berarti hitung nilai min dan max untuk setiap dataset secara independen menggunakan fit\_transform(). Ini akan memungkinkan setiap dataset untuk diskalakan secara terpisah, yang bukan pendekatan yang benar untuk menjaga integritas model dan generalisasi."

Kalau untuk metode normalisasi Min-Max Scaling, Z-Score (Standard Scaling) dan Decimal Scaling, apakah logikanya tetap sama dengan:

Normalisasi pada training: Menghitung statistik (mean, standard deviation, atau min-max) berdasarkan data training dan melakukan normalisasi pada data training. Normalisasi pada testing: Tidak menghitung statistik baru, tetapi hanya menerapkan transformasi yang sudah dihitung pada data training ke data testing.

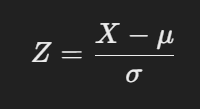
Ya, logika untuk normalisasi dengan metode **Min-Max Scaling**, **Z-Score (Standard Scaling)**, dan **Decimal Scaling** tetap sama. Prinsip utama adalah Anda **hanya menghitung statistik** (misalnya, nilai min, max, mean, atau standar deviasi) berdasarkan **data training** dan **menerapkan transformasi yang sama** pada data testing, tanpa menghitung statistik baru untuk data testing. Berikut penjelasan lebih lanjut untuk masing-masing metode:

**1. Min-Max Scaling**

* **Statistik yang dihitung:** Min dan Max dari fitur dalam dataset training.
* **Transformasi pada data training:** Hitung nilai min dan max, kemudian normalisasi data training sehingga berada dalam rentang tertentu (misalnya, [0, 1]).
* **Transformasi pada data testing:** Gunakan nilai min dan max yang dihitung dari data training untuk menerapkan normalisasi pada data testing, **tanpa menghitung ulang min dan max** dari data testing.

**2. Z-Score (Standard Scaling)**

* **Statistik yang dihitung:** Mean dan Standard Deviation dari fitur dalam dataset training.
* **Transformasi pada data training:** Hitung mean dan standard deviation, kemudian ubah data training dengan rumus:



di mana μ adalah mean dan σ adalah standard deviation.

* **Transformasi pada data testing:** Gunakan nilai mean dan standard deviation dari data training untuk menerapkan normalisasi pada data testing, **tanpa menghitung ulang mean dan standard deviation** pada data testing.

**3. Decimal Scaling**

* **Statistik yang dihitung:** Tidak ada statistik seperti min, max, atau mean yang dihitung. Sebaliknya, **nilai j** dihitung berdasarkan **nilai terbesar** dalam fitur di dataset training. Menentukan nilai j berdasarkan data training, di mana j adalah jumlah digit yang diperlukan untuk mengubah nilai fitur ke dalam rentang tertentu.
* **Transformasi pada data training:** Tentukan nilai j dengan melihat nilai terbesar dalam data training, kemudian lakukan scaling dengan membagi nilai fitur dengan 10j.
* **Transformasi pada data testing:** Gunakan nilai j yang dihitung dari data training untuk mengubah data testing, **tanpa menghitung ulang j** berdasarkan data testing.

**Kesimpulan:**

**Logika umum untuk semua metode normalisasi** (Min-Max, Z-Score, Decimal Scaling) tetap sama:

* **Hitung statistik (misalnya, mean, min, max, atau standar deviasi) hanya berdasarkan data training.**
* **Terapkan transformasi yang sama pada data testing menggunakan statistik yang dihitung dari data training.**
* **Jangan menghitung statistik baru atau transformasi pada data testing.**

Dengan cara ini, Anda memastikan bahwa data testing tetap "tidak terlihat" selama training dan evaluasi model yang lebih realistis dan terhindar dari kebocoran data.

yasudah kalau begitu tolong buatkan promt untuk metode normalisasi Min-Max Scaling, Z-Score (Standard Scaling), dan Decimal Scaling. dengan cara normalisasi:

Normalisasi pada training: Menghitung statistik berdasarkan data training dan melakukan normalisasi pada data training. Normalisasi pada testing: Tidak menghitung statistik baru, tetapi hanya menerapkan transformasi yang sudah dihitung pada data training ke data testing.

1. **Min-Max Scaling**

"Lakukan normalisasi menggunakan Min-Max Scaling pada dataset training dan dataset testing. Pada tahap pertama, hitung nilai min dan max untuk setiap fitur dalam dataset training. Kemudian, normalisasi dataset training dengan transformasi: X\_scaled = (X - min) / (max - min). Setelah itu, terapkan transformasi yang sama pada dataset testing dengan menggunakan nilai min dan max yang dihitung dari dataset training, tanpa menghitung ulang min dan max untuk data testing."

"Lakukan normalisasi menggunakan Min-Max Scaling pada dataset pelatihan dan dataset pengujian. Pada tahap pertama, hitung nilai minimum dan maksimum untuk setiap fitur dalam dataset pelatihan. Kemudian, lakukan normalisasi pada dataset pelatihan dengan menggunakan nilai minimum dan maksimum yang dihitung tersebut. Setelah itu, terapkan transformasi yang sama pada dataset pengujian dengan menggunakan nilai minimum dan maksimum yang telah dihitung dari dataset pelatihan, tanpa menghitung ulang nilai untuk dataset pengujian."

2. **Z-Score (Standard Scaling)**

"Lakukan normalisasi menggunakan Z-Score (Standard Scaling) pada dataset training dan dataset testing. Pada tahap pertama, hitung nilai mean dan standard deviation untuk setiap fitur dalam dataset training. Kemudian, normalisasi dataset training dengan transformasi: X\_scaled = (X - mean) / standard deviation. Setelah itu, terapkan transformasi yang sama pada dataset testing dengan menggunakan nilai mean dan standard deviation yang dihitung dari dataset training, tanpa menghitung ulang nilai mean dan standard deviation untuk data testing."

"Lakukan normalisasi menggunakan Z-Score (Standard Scaling) pada dataset pelatihan dan dataset pengujian. Pada tahap pertama, hitung nilai mean dan standar deviasi untuk setiap fitur dalam dataset pelatihan. Kemudian, lakukan normalisasi pada dataset pelatihan dengan menggunakan nilai mean dan standar deviasi yang dihitung tersebut. Setelah itu, terapkan transformasi yang sama pada dataset pengujian dengan menggunakan nilai mean dan standar deviasi yang dihitung dari dataset pelatihan, tanpa menghitung ulang nilai untuk dataset pengujian."

3. **Decimal Scaling**

"Lakukan normalisasi menggunakan Decimal Scaling pada dataset training dan dataset testing. Pada tahap pertama, tentukan nilai j berdasarkan dataset training, di mana j adalah jumlah digit yang diperlukan untuk mengubah nilai fitur ke dalam rentang yang diinginkan. Kemudian, normalisasi dataset training dengan transformasi: X\_scaled = X / 10^j. Setelah itu, terapkan transformasi yang sama pada dataset testing dengan menggunakan nilai j yang dihitung dari dataset training, tanpa menghitung ulang j untuk data testing."

"Lakukan normalisasi menggunakan Decimal Scaling pada dataset pelatihan dan dataset pengujian. Pada tahap pertama, tentukan faktor skala berdasarkan dataset pelatihan, yang diperlukan untuk mengubah nilai fitur ke dalam rentang yang diinginkan. Kemudian, lakukan normalisasi pada dataset pelatihan dengan menggunakan faktor skala yang telah ditentukan. Setelah itu, terapkan transformasi yang sama pada dataset pengujian dengan menggunakan faktor skala yang dihitung dari dataset pelatihan, tanpa menghitung ulang faktor skala untuk dataset pengujian."