**Jelaskan bagaimana teknik 'Data Blending' dan 'Data Munging' berperan dalam proses Data Science dan berikan contoh spesifik bagaimana keduanya diterapkan dalam proyek nyata.**

Data Munging :  (Proses atau step untuk cleaning data dan persiapan data atau pengertian lainnya adalah penyimpanan data atau perselisihan data mengacu pada proses pembersihan dan transformasi data mentah ke dalam format yang diinginkan, biasanya untuk memfasilitasi analisis atau visualisasi lebih lanjut. Pengolahan data dapat dilakukan dengan Python atau R, tetapi juga dapat dilakukan dengan program spreadsheet.

Contoh : Urutan dalam kehidupan nyata sebagai DS melakukan : Data Pre- Processing  --→ Data Enriching --→ Data Validation

Keuntungan : Data munging will help to prepare and manipulate data before data scientists go through rigorous analysis. There are many benefits of data munging such as

* It increases efficiency,
* Final results are more accurate.
* Access to quality data.
* This leads to data driven decisions.

Data Blending : Pencampuran data adalah proses menggabungkan data dari berbagai sumber menjadi kumpulan data yang berfungsi. Proses ini mendapatkan perhatian di kalangan analis dan perusahaan analitik karena merupakan metode cepat dan mudah yang digunakan untuk mengekstraksi nilai dari berbagai sumber data.

Contoh : Gabungkan data dari berbagai sumber dan sesuaikan setiap gabungan berdasarkan dimensi umum untuk memastikan pencampuran data berjalan lancar. Pikirkan tentang pandangan campuran yang diinginkan dan hanya sertakan data yang penting untuk menjawab pertanyaan yang diajukan serta bidang apa pun yang dapat memberikan konteks tambahan pada jawaban tersebut ketika analisis ditekankan. Kumpulan data yang dihasilkan harus mudah dipahami dan dijelaskan kepada pemangku kepentingan. Lingkari kembali langkah ini untuk menyertakan atau menghapus data dari alur kerja dan menyusun analisis lebih lanjut.

**Jelaskan bagaimana 'Data Structure' di Python (seperti list, dictionary, sets) dapat mempengaruhi performa komputasi dalam proyek data science.**

Struktur data merupakan suatu *tool*yang digunakan oleh para *data scientist*untuk memperbarui, mengelola, dan bahkan mengindeks layanan internet secara efisien. Dengan kata lain, struktur data adalah sebuah cara untuk mengatur data dalam format khusus di komputer sehingga informasi dapat diproses dengan cepat dan efektif.

Di samping itu, struktur data pun dapat dijadikan dasar untuk banyak algoritma. Ini karena struktur data dapat memfilter dan mengurutkan data dalam skala besar. Untuk bisa bekerja, struktur data ditunjang oleh tiga aspek utama berikut ini:

1. ***Memory address*:** Elemen data mentah tetap dari fitur atau fungsi yang diinginkan.
2. ***Pointer*:**Alat referensi yang merujuk pada lokasi alamat memori.
3. ***Procedure***: Suatu kode tertulis untuk membuat fungsi berbeda dalam struktur, baik secara manual atau otomatis.

Ketiganya amat penting dalam menunjang kinerja struktur data karena cara mereka diwakili dalam *database* atau situs web tertentu, menentukan berbagai jenis struktur yang dapat dimasukkan ke dalam *database*.

**Skewness and Kurtosis' memberikan informasi tentang bentuk distribusi data dan implikasinya terhadap analisis data.**

Kemencengan atau kecondongan (skewness) adalah tingkat ketidaksimetrisan atau kejauhan simetri dari sebuah distribusi. Skewness diartikan sebagai kemiringan distribusi data. Sebuah distribusi yang tidak simetris akan memiliki rata-rata, median, dan modus yang tidak sama besarnya sehingga distribusi akan terkonsentrasi pada salah satu sisi dan kurvanya akan menceng. Ukuran kemiringan kurva adalah derajat atau ukuran dari ketidaksimetrian suatu distribusi data. Nilai skewness (ukuran kemiringan) menunjukkan data normal ketika nilai-nilai tersebut berada di antara rentang nilai -2 sampai dengan 2. Kurva positif apabila rata-rata hitung > modus/median. Kurva negative apabila rata-rata hitung < modus/media. Jika distribusi memiliki ekor yang lebih panjang ke kanan daripada yang ke kiri maka distribusi disebut menceng ke kanan atau memiliki kemencengan positif. Sebaliknya, jika distribusi memiliki ekor yang lebih panjang ke kiri daripada yang ke kanan maka distribusi disebut menceng ke kiri atau memiliki kemencengan negatif. Berikut ini gambar kurva dari distribusi yang menceng ke kanan (menceng positif) dan menceng ke kiri

Kriteria untuk mengetahui model distribusi dari koefisien kemiringan :

* Jika koefisien kemiringan <nol, maka bentuk distribusinya negative (ekor bagian kiri lebih Panjang).
* Jika koefisien kemiringan = nol, maka bentuk distribusinya simetrik.
* Jika koefisien kemiringan > nol, maka bentuk distribusinya positif (ekor bagian kanan lebih Panjang).

Sebagai contoh Nilai *skewness* pada variabel keamanan sistem menunjukkan data normal ketika nilai-nilai tersebut berada di antara rentang nilai -2 sampai dengan 2. Nilai *skewness* pada indikator keamanan system (K) misal K1 adalah -0.646, K2 adalah -0.373 dan K3 adalah -0.455. Semua nilai *skewness* pada indikator tersebut berada pada rentang nilai -2 sampai dengan 2, **sehingga dapat disimpulkan bahwa data tersebut normal**

**Jelaskan pentingnya dan tantangan dalam 'Data Transformation' khususnya dalam konteks normalisasi dan standarisasi data.**

Pentingnya ‘Data Transformation’ :

Data transformation digunakan untuk mengubah data dalam bentuk yang sesuai dalam proses data mining. Beberapa teknik untuk data transformation adalah normalization, pemilihan attribute, dan discretization. Normalization dilakukan untuk menskalakan nilai data dalam rentang nilai tertentu, misalnya -1 sampai 1 atau 0 sampai 1. Teknik kedua adalah pemilihan atribut. Pemilihan atribute merupakan proses pemilihan atribut yang diberikan untuk proses data mining. Terakhir adalah teknik discretization. Teknik ini dilakukan untuk mengganti raw value pada atribut numerik dengan nilai interval

Tantangan dalam ‘Data Transformation’ :

Harus melakukan data celaning terlebih dahulu sampai data tersebut layak untuk di lanjutkan misalnya seperti mengetahui Mean Q1 Q2 dan Q3  apakah terjadi gap yang sangat jauh angkanya jika tidak maka layak dan mengetahui sebaran data menggunakan scatter plot serta melihat dari hasil histogram bagaimana sebaran dan outlier yang terjadi.

**Bagaimana 'Categorical Data Encoding' seperti One-Hot Encoding dan Label Encoding dapat mempengaruhi model machine learning dan kapan masing-masing harus digunakan.**

Categorical data atau data kategorikal adalah data yang bersifat karakteristik dan umumnya direpresentasikan dalam kata atau bilangan yang tidak bisa diukur. Categorical data dibagi menjadi dua jenis, yaitu nominal dan ordinal.

Data nominal: data yang tidak memiliki urutan atau ranking. Contohnya jenis kelamin, warna mata, atau jenis produk.

Data ordinal: data yang memiliki urutan atau ranking. Contohnya tingkat pendidikan, kepuasan pelanggan, atau penilaian sekolah.

Machine learning tidak mampu mengolah data yang bersifat kategorikal. Sehingga perlu adanya proses manipulasi data sehingga seolah-olah terbaca sebagai data numerik yang kemudian dapat digunakan untuk proses analisis data. Tidak hanya bertujuan untuk memudahkan proses memodelkan machine learning, tapi dengan adanya encoding juga membantu mengurangi dimensi data, sehingga mempercepat waktu pemrosesan dan penyimpanan data.

**One Hot Encoding**

Tipe encoding yang pertama adalah one hot encoding. Tipe ini bisa digunakan pada data nominal atau yang tidak memiliki urutan. Cara encodingnya adalah mengubah categorical data menjadi sebuah vektor biner dengan nilai 1 pada kategori yang sesuai dan 0 untuk kategori lainnya.

**Label Encoding**

Jika pada one hot encoding kita menggunakan angka 0 dan 1, maka pada label encoding kita gunakan angka 0 sampai banyaknya data minus 1. Contohnya, dalam variabel ‘Kota’ berisi Bandung, Jogja, Kediri, dan Surabaya. Maka akan digunakan label 0, 1, 2, dan 3. Untuk lebih jelasnya perhatikan contoh berikut.

**Jelaskan perbedaan antara 'Parametric' dan 'Non-Parametric' tests dalam hypothesis testing, dan berikan contoh kasus di mana setiap metode lebih tepat digunakan.**

Statistik parametrik adalah uji hipotesis yang menguji perbedaan rata-rata pada populasi. Metode yang sering digunakan dalam statistik parametrik adalah uji-t yang didasarkan pada nilai student-t statistics. Uji t bertumpu pada asumsi bahwa data berdistribusi normal dan rata-rata data diketahui. Pada uji ini, varians populasi dihitung untuk mencari sampel dari populasi tersebut. Populasi diperkirakan dengan bantuan skala interval dan variabel yang yang akan diuji hipotesisnya.

Statistik non parametrik adalah uji yang tidak membutuhkan asumsi parameter apapun untuk populasi yang diuji atau dalam bahasa sederhana uji ini tidak bergantung pada populasi. Dalam uji statistik non parametrik, tidak ada parameter yang digunakan dan tidak ada distribusi yang harus diketahui. Hal ini menyebabkan uji statistik non parametrik juga disebut sebagai metode bebas distribusi.

Jika sebuah uji memerlukan asumsi spesifik mengenai parameter populasi, maka uji tersebut dikenal dengan uji statistik parametrik sedangkan jika uji tersebut tidak membutuhkan asumsi spesifik seperti distribusi data, maka uji tersebut dikenal dengan uji statistik non parametrik. Dalam uji parametrik, diasumsikan bahwa pengukuran variabel dilakukan pada tingkat internal dan rasio, sedangkan pada uji non parametrik variabel diukur dengan skala nominal dan ordinal. Secara umum, ukuran central tendency dalam uji parametrik adalah rata-rata, sedangkan dalam uji statistik non parametrik adalah median. Terakhir, dalam uji statistik parametrik terdapat informasi lengkap mengenai populasi sedangkan pada uji statistik non parametrik tidak ada informasi mengenai populasi

Contoh Parametrik :

Analisis Korelasi pada uji statistik parametrik digunakan untuk menguji hubungan antar variabel. Hubungan yang dimaksud disini adalah keeratan hubungan antara dua variabel atau lebih tanpa memperhatikan ada tidaknya hubungan kausal di antara variabel-variabel yang dianalisis. Analisis korelasi yang digunakan untuk menguji data yang bersifat parametrik sering juga disebut dengan analisis Pearson. Metode analisis korelasi Pearson merupakan metode pengujian yang pada dasarnya dilakukan terhadap suatu variabel pengukuran yang menyajikan data-data yang bersifat kuantitatif

Contoh Non-Parametrik (chi square test)

Uji Chi Square pada uji statistik non parametric atau dikenal juga sebagai uji Kai Kuadrat, adalah salah satu cara yang digunakan untuk menyampaikan atau menunjukkan keberadaan hubungan (ada atau tidaknya) antara variabel yang diteliti. Misalkan sebagai peneliti, kita hendak melakukan uji terhadap perilaku mahasiswa. Karakter yang akan diuji adalah perilaku mahasiswa yang dikategorikan menjadi dua kategori. Kategori pertama yaitu mahasiswa yang mendukung program kampus dan kedua adalah yang acuh terhadap program kampus. Kondisi tersebut memungkinkan kita untuk melakukan uji hipotesis mengenai perbedaan perilaku mahasiswa tersebut dilihat dari frekuensinya.

**Apa yang dimaksud dengan p-value dalam konteks hypothesis testing dan bagaimana interpretasinya?**

Dalam statistika, istilah *p value* atau nilai P sangat sering disebut terutama ketika melakukan uji hipotesis.

Dikutip dari Investopedia, *p value***adalah pengukuran statistika untuk memvalidasi hipotesis terhadap data yang diamati.***P value* mengukur kemungkinan hipotesis sesuai dengan pengujian yang diamati.

*P value* dinyatakan sebagai desimal, tapi bisa diubah dalam bentuk persentase agar lebih mudah membaca [probabilitas](https://revou.co/kosakata/probabilitas) hipotesisnya.

Misalnya, didapatkan *p value* suatu penelitian sebesar 0,0254. Jika diubah dalam bentuk persentase, maka probabilitasnya adalah 2,54%.

Dalam statistika, istilah *p value* atau nilai P sangat sering disebut terutama ketika melakukan uji hipotesis.

Dikutip dari Investopedia, *p value***adalah pengukuran statistika untuk memvalidasi hipotesis terhadap data yang diamati.***P value* mengukur kemungkinan hipotesis sesuai dengan pengujian yang diamati.

*P value* dinyatakan sebagai desimal, tapi bisa diubah dalam bentuk persentase agar lebih mudah membaca [probabilitas](https://revou.co/kosakata/probabilitas) hipotesisnya.

Misalnya, didapatkan *p value* suatu penelitian sebesar 0,0254. Jika diubah dalam bentuk persentase, maka probabilitasnya adalah 2,54%.

Contoh dalam intrepertasinya adalah :

Salah satu contoh sederhana *p value* adalah untuk menentukan probabilitas bagian koin yang muncul selama proses pelemparan koin.

Setiap koin memiliki dua bagian, yaitu kepala dan ekor. *P value* menentukan probabilitas jumlah kepala dan ekor yang muncul dalam sepuluh kali lemparan.

Contoh kegunaan *p value* juga bisa terlihat dalam bidang investasi saham.

Misalnya, ASEP percaya kinerja portofolionya setara dengan indeks S&P 500, indeks acuan penting di AS yang terdiri dari 500 bursa saham terbesar di Brazil. Untuk menentukan hal tersebut, ia melakukan tes *two-tailed.*

Hipotesis awalnya (H0) adalah *return* portofolionya setara dengan *return* S&P 500. Sedangkan hipotesis alternatifnya (Ha) dalah *return* portofolionya tidak setara dengan *return* S&P 500.

Setelah dilakukan uji hipotesis, didapatkan hasil *p value* sebesar 0,01 atau kurang dari 0,05. Hal ini menunjukkan bahwa *return* portofolionya tidak setara dengan *return* S7P 500 (H0 ditolak, Ha diterima).

**Berikan contoh kasus di mana 'WITH RECURSIVE' query bisa berguna dalam SQL dan jelaskan bagaimana query tersebut berfungsi.**

Untuk membuat kueri SQL rekursif, Anda perlu mendefinisikan CTE dengan dua bagian: anggota jangkar dan anggota rekursif. Anggota jangkar menetapkan titik awal untuk rekursi, sedangkan anggota rekursif menentukan hubungan antara baris induk dan baris anak. Berikut struktur dasar CTE rekursif:

WITH RECURSIVE cte\_name (column\_names) AS

( -- Anchor member

 SELECT ...

UNION ALL

 -- Recursive member

 SELECT ... )

SELECT \* FROM cte\_name;

Let's break down the components of this example:

1. **WITH RECURSIVE**: The **RECURSIVE** keyword indicates that the CTE is recursive.
2. **cte\_name**: The name of the CTE.
3. **(column\_names)**: An optional list of column names for the CTE.
4. **SELECT ...**: The anchor member, which selects the initial rows for the recursion.
5. **UNION ALL**: Combines the results of the anchor and recursive members.
6. **SELECT ...**: The recursive member, which defines the relationship between parent and child rows.

|  |  |  |  |
| --- | --- | --- | --- |
| Employee ID | First Name | last Name | Manager ID |
| 1 | Alice | Johnson | Null |
| 2 | Bob | Smith | 1 |
| 3 | Carol | Brown | 2 |
| 4 | Dave | Miller | 1 |
| 5 | Emily | White | 2 |

In this table, each employee has an **employee\_id**, **employee\_name**, and `manager\_id(continuing from where we left off…)

Here, **manager\_id** is the **employee\_id** of the employee's manager. For example, Jane Doe (employee\_id: 2) is John Doe's (employee\_id: 1) subordinate because Jane's **manager\_id** is 1.

Suppose you need to find out the complete hierarchy for each employee, i.e., who their manager is, who their manager's manager is, and so on. You can use a recursive query to solve this problem.

Here's the SQL recursive query that does this:

WITH RECURSIVE employee\_hierarchy AS (

    -- Base case

    SELECT employee\_id, employee\_name, manager\_id

FROM employees

    WHERE manager\_id IS NULL

    UNION ALL

    -- Recursive case

    SELECT e.employee\_id, e.employee\_name, e.manager\_id

    FROM employees e

    INNER JOIN employee\_hierarchy eh ON e.manager\_id = eh.employee\_id

)

SELECT \* FROM employee\_hierarchy;

WITH RECURSIVE employee\_hierarchy AS (

-- Base case

SELECT employee\_id, employee\_name, manager\_id

FROM employees

WHERE manager\_id IS NULL

UNION ALL

-- Recursive case

SELECT e.employee\_id, e.employee\_name, e.manager\_id

FROM employees e

INNER JOIN employee\_hierarchy eh ON e.manager\_id = eh.employee\_id

)

SELECT \* FROM employee\_hierarchy;

**Jelaskan bagaimana 'Window Functions' seperti ROW\_NUMBER(), RANK(), DENSE\_RANK() berbeda satu sama lain dan berikan contoh kasus penggunaannya.**

Jawaban :

Before we explore these SQL RANK functions, let’s prepare sample data. In this sample data, we have exam results for three students in Maths, Science and English subjects.

CREATE TABLE ExamResult

(StudentName VARCHAR(70),

Subject     VARCHAR(20),

Marks       INT

);

INSERT INTO ExamResult

VALUES

('Lily',

'Maths',

65

);

INSERT INTO ExamResult

VALUES

('Lily',

'Science',

80

);

INSERT INTO ExamResult

VALUES

('Lily',

'english',

70

);

INSERT INTO ExamResult

VALUES

('Isabella',

'Maths',

50

);

INSERT INTO ExamResult

VALUES

('Isabella',

'Science',

70

);

INSERT INTO ExamResult

VALUES

('Isabella',

'english',

90

);

INSERT INTO ExamResult

VALUES

('Olivia',

'Maths',

55

);

INSERT INTO ExamResult

VALUES

('Olivia',

'Science',

60

);

INSERT INTO ExamResult

VALUES

('Olivia',

'english',

89

hasil dari SQL akan seperti pada tabel berikut :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student Name | Subject | Marks |
| 1 | Lily | Maths | 65 |
| 2 | Lily | Science | 80 |
| 3 | Lily | english | 70 |
| 4 | Isabella | Maths | 50 |
| 5 | Isabella | Science | 70 |
| 6 | Isabella | english | 90 |
| 7 | Olivia | Maths | 55 |
| 8 | Olivia | Science | 60 |
| 9 | Olivia | english | 89 |

**ROW\_Number() SQL RANK function**

We use ROW\_Number() SQL RANK function to get a unique sequential number for each row in the specified data. It gives the rank one for the first row and then increments the value by one for each row. We get different ranks for the row having similar values as well.

Execute the following query to get a rank for students as per their marks.

SELECT Studentname,

       Subject,

       Marks,

       ROW\_NUMBER() OVER(ORDER BY Marks) RowNumber

FROM ExamResult;

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student Name | Subject | Marks |
| 1 | Isabella | Maths | 50 |
| 2 | Olivia | Maths | 55 |
| 3 | Olivia | Science | 60 |
| 4 | lily | Maths | 65 |
| 5 | Isabella | Science | 70 |
| 6 | lily | english | 70 |
| 7 | lily | Science | 80 |
| 8 | Olivia | english | 89 |
| 9 | Isabella | english | 90 |

By default, it sorts the data in ascending order and starts assigning ranks for each row. In the above screenshot, we get ROW number 1 for marks 50.

We can specify descending order with Order By clause, and it changes the RANK accordingl

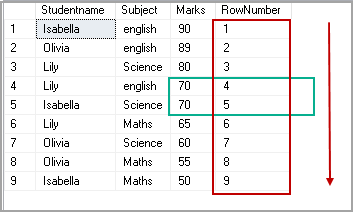
SELECT Studentname,

       Subject,

       Marks,

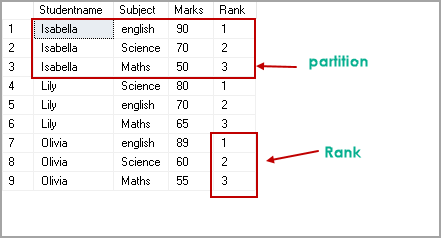
       ROW\_NUMBER() OVER(ORDER BY Marks desc) RowNumber

FROM ExamResult;



**RANK() SQL RANK Function**

We use RANK() SQL Rank function to specify rank for each row in the result set. We have student results for three subjects. We want to rank the result of students as per their marks in the subjects. For example, in the following screenshot, student Isabella got the highest marks in English subject and lowest marks in Maths subject. As per the marks, Isabella gets the first rank in English and 3rd place in Maths subject.



Execute the following query to get this result set. In this query, you can note the following things:

* We use PARTITION **BY Studentname** clause to perform calculations on each student group
* Each subset should get rank as per their Marks in descending order
* The result set uses Order By clause to sort results on Studentname and their rank

SELECT Studentname,

       Subject,

       Marks,

       RANK() OVER(PARTITION BY Studentname ORDER BY Marks DESC) Rank

FROM ExamResult

ORDER BY Studentname,

         Rank;

Let’s execute the following query of SQL Rank function and look at the result set. In this query, we did not specify SQL PARTITION By clause to divide the data into a smaller subset. We use SQL Rank function with over clause on Marks clause ( in descending order) to get ranks for respective rows

SELECT Studentname,

       Subject,

       Marks,

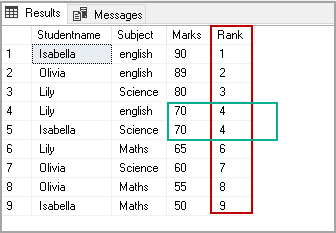
       RANK() OVER(ORDER BY Marks DESC) Rank

FROM ExamResult

ORDER BY Rank;

In the output, we can see each student get rank as per their marks irrespective of the specific subject. For example, the highest and lowest marks in the complete result set are 90 and 50 respectively. In the result set, the highest mark gets RANK 1, and the lowest mark gets RANK 9.

If two students get the same marks (in our example, ROW numbers 4 and 5), their ranks are also the same.



**DENSE\_RANK() SQL RANK function**

We use DENSE\_RANK() function to specify a unique rank number within the partition as per the specified column value. It is similar to the Rank function with a small difference.

In the SQL RANK function DENSE\_RANK(), if we have duplicate values, SQL assigns different ranks to those rows as well. Ideally, we should get the same rank for duplicate or similar values.

Let’s execute the following query with the DENSE\_RANK() function

SELECT Studentname,

       Subject,

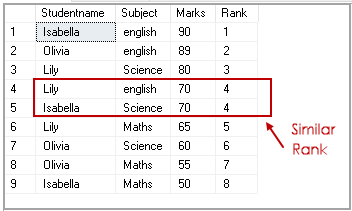
       Marks,

       DENSE\_RANK() OVER(ORDER BY Marks DESC) Rank

FROM ExamResult

ORDER BY Rank;

In the output, you can see we have the same rank for both Lily and Isabella who scored 70 marks.



Let’s see the difference between RANK() and DENSE\_RANK() SQL Rank function with the following query

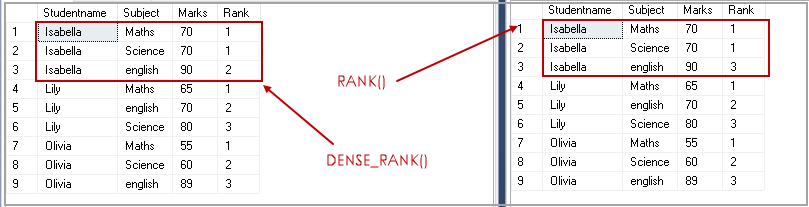
* Query 1

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | SELECT Studentname,         Subject,         Marks,         RANK() OVER(PARTITION BY StudentName ORDER BY Marks ) Rank  FROM ExamResult  ORDER BY Studentname,           Rank; |

* Query 2

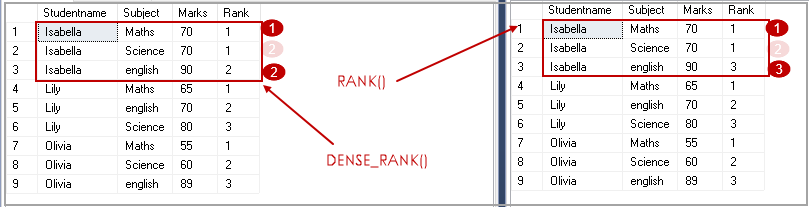
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | SELECT Studentname,         Subject,         Marks,         DENSE\_RANK() OVER(PARTITION BY StudentName ORDER BY Marks ) Rank  FROM ExamResult  ORDER BY Studentname,           Rank; |

In the output, you can see a gap in the rank function output within a partition. We do not have any gap in the DENSE\_RANK function.



In the following screenshot, you can see that Isabella has similar numbers in the two subjects. A rank function assigns rank 1 for similar values however, internally ignores rank two, and the next row gets rank three.

In the Dense\_Rank function, it maintains the rank and does not give any gap for the values.

 **Dalam SQL, bagaimana 'Window Functions' dapat digunakan bersamaan dengan 'Subqueries' untuk menghasilkan analisis data yang kompleks, dan berikan contoh penggunaan query pada kasus tertentu** Windows function:

A window function performs a calculation across a set of table rows that are related to the current row. In this example, we will calculate the running total for each salesperson's sales.

SELECT salesperson\_id, sale\_date, sale\_amount,

                      SUM(sale\_amount) OVER (PARTITION BY salesperson\_id

ORDER BY sale\_date) as running\_total

FROM sales;

A screenshot of a computer

Description automatically generated

Subquery :

A subquery is a query embedded within another query, often used to filter or calculate aggregated data. In this example, we want to calculate the total sales for each salesperson and return the rows where the total sales exceed a certain threshold.

SELECT s1.salesperson\_id, SUM(s1.sale\_amount) as total\_sale

FROM sales s1

WHERE s1.salesperson\_id IN (

SELECT s2.salesperson\_id

 FROM sales s2    GROUP BY s2.salesperson\_id

HAVING SUM(s2.sale\_amount) > 5000

)

GROUP BY s1.salesperson\_id;s

**Bagaimana cara visualisasi data bivariat dengan Python dan apa tujuannya?** Bivariate analysis digunakan untuk untuk menganalisa 2 variables dan menemukan sebuah relasi. Analisis bivariate juga merupakan salah satu cara untuk menggunakan koefisien korelasi dalam rangka menemukan apakah dua variabel memiliki relasi atau tidak.

Cara atau contoh Bivariat dengan Python :

dalam menganalisa dua buah data adalah dengan menggunakan hexbin plot. Hexbin plot berguna untuk merepresentasikan hubungan dua buah variabel numerik ketika memiliki titik-titik data. Jika sebelumnya kita dihadapi terhadap masalah overlapping, solusinya adalah dengan mengcounting jumlah data yang overlap pada titik-titik hexagonal tersebut. Warna-warna tersebut merepresentasikan jumlah titik-titik yang overlap. Jalankanlah program berikut:

A white rectangular sign with black text

Description automatically generated

A green and white hexagons

Description automatically generated

**Jelaskan bagaimana 'Data Binding' dalam pengembangan dashboard mempengaruhi interaktivitas dan visualisasi data**

Data Bindings are a way of keeping a component's properties in sync with the underlying data with minimal effort. A Data Binding associates a property of a component with a single Python expression.

pada penggunaannya Data Binding sangat berpengeruh terhadap update real time sebuah data dan dashboard yang dinamis atau berubah-berubah sehingga pola atau rumus dasar pada sebuah data tidak berubah hanya data in dan out nya saja yang berubah sesuai dengan kondisi realtime

**Jelaskan bagaimana 'Principal Component Analysis (PCA)' dapat digunakan untuk memahami struktur data. Apa kelebihan dan keterbatasan metode ini?**

*Principal component analysis* (PCA) adalah suatu teknik analisis statistik multivariat. Bisa dibilang, inilah teknik analisis statistik yang paling populer sekarang. Biasanya, PCA digunakan dalam bidang pengenalan pola serta pemrosesan sinyal.

PCA pada dasarnya merupakan dasar dari analisis data multivariat yang menerapkan metode proyeksi. Teknik analisis ini biasanya digunakan untuk meringkas tabel data multivariat dalam skala besar hingga bisa dijadikan kumpulan variabel yang lebih kecil atau indeks ringkasan. Dari situ, kemudian variabel dianalisis untuk mengetahui tren tertentu, klaster variabel, hingga *outlier.*

**Kelebihan**:

1. Mengurangi dimensi variabel: PCA membantu mengurangi dimensi variabel dalam dataset yang kompleks, sehingga dapat mempercepat analisis data dan membuat hasil yang lebih mudah dipahami.
2. Meningkatkan efisiensi: Dengan menghilangkan variabel yang tidak penting dan mengidentifikasi hubungan antara variabel, PCA dapat meningkatkan efisiensi dan akurasi analisis data.
3. Meningkatkan pemahaman pola dalam data: PCA dapat membantu meningkatkan pemahaman pola dalam data yang kompleks, sehingga memungkinkan pengambilan keputusan yang lebih tepat dan efektif.
4. Dapat digunakan untuk berbagai jenis data: PCA dapat digunakan untuk berbagai jenis data, termasuk data numerik, data kategori, dan data ordinal

**Kekurangan**:

1. Ketergantungan pada data: PCA sangat bergantung pada data yang dianalisis dan mungkin tidak memberikan hasil yang optimal jika data tidak berdistribusi normal.
2. Interpretasi yang sulit: Hasil PCA dapat sulit diinterpretasikan karena komponen utama tidak selalu memiliki arti yang jelas dalam konteks data asli.
3. Kerugian informasi: PCA dapat menyebabkan kerugian informasi yang signifikan jika komponen utama yang dihasilkan tidak dapat menjelaskan sebagian besar variasi dalam data.
4. Sensitif terhadap skala: PCA sensitif terhadap skala variabel yang diukur, sehingga perlu dilakukan normalisasi skala variabel untuk menghindari efek yang merugikan.

Dalam prakteknya, kelebihan dan kekurangan PCA harus dipertimbangkan dengan cermat untuk memutuskan apakah teknik ini tepat untuk digunakan pada dataset tertentu.

**Jelaskan bagaimana metode Imputation dapat digunakan dalam EDA untuk mengatasi missing values dan sebutkan setidaknya dua teknik imputation yang berbeda.**

Ada banyak cara untuk memperhitungkan nilai pengganti data, seperti :

1. Nilai konstanta yang merupakan bagian dari kumpulan nilai yang mungkin dari variabel itu, seperti 0, berbeda dari semua nilai lainnya
2. Nilai rata-rata, median, atau mode untuk kolom
3. Nilai yang diperkirakan oleh model prediktif lain
4. Beberapa Imputasi lain

contoh imputasi dengan menggunakan mean

A close up of a text

Description automatically generated

atau dapat menggunakan

data\_full ['BuildingArea'].fillna)data\_full['BuildingArea'].mode(),

implace=True)

data\_full.head(5)

**Jelaskan bagaimana teknik 'Feature Selection' menggunakan metode statistik dapat digunakan dalam EDA untuk meningkatkan efektivitas model Machine Learning, dan berikan contoh metode tersebut.**

Secara garis besar ada dua jenis feature selection, yaitu supervised dan unsupervised. Metode supervised terdiri dari metode wrapper, filter dan intrinsik/embedded.

Perbedaan antara supervised dan unsupervised feature selection mirip dengan perbedaan antara supervised dan unsupervised learning, yaitu pada penggunaan variabel output / target.

1. **Unsupervised feature selection** : tidak menggunakan variabel output dalam melakukan seleksi, misalnya memilih antara fitur-fitur yang redundan. Proses ini hanya membandingkan antar variabel input saja.
2. **Supervised feature selection** : menggunakan variabel output dalam seleksi, yaitu misalnya membuang variabel yang tidak relevan terhadap variabel output.

Metode pada supervised feature selection dapat dibagi lagi menjadi 3 jenis, yaitu: filter, wrapper, dan embedded/intrinsik

**1. Metode Filter**

Metode filter mengevaluasi setiap fitur secara bebas dari classifier kemudian memberikan peringkat pada fitur setelah mengevaluasi dan mengambil yang unggul.

Metode filter menerapkan ukuran statistik untuk menetapkan skor untuk setiap fitur. Fitur-fitur tersebut diberi peringkat berdasarkan skor dan dipilih untuk disimpan atau dihapus dari dataset. Metode ini sering bersifat univariat dan mempertimbangkan fitur secara mandiri, atau berkaitan dengan variabel dependen.

|  |
| --- |
|  |
| *Sumber: analyticsvidhya.com* |

Metode filter bergantung pada keunikan umum dari data yang akan dievaluasi dan memilih subset fitur. Metode filter menggunakan kriteria penilaian yang mencakup jarak, informasi, ketergantungan, dan konsistensi. Metode filter menggunakan kriteria utama teknik pemeringkatan dan menggunakan urutan peringkat untuk pemilihan variabel.

Alasan untuk menggunakan metode pemeringkatan adalah kesederhanaan, menghasilkan fitur yang sangat baik dan relevan. Metode pemeringkatan akan menyaring fitur yang tidak relevan sebelum proses klasifikasi dimulai.

Metode filter umumnya digunakan sebagai langkah preprocessing data. Penentuan dan pemilihan fitur tidak tergantung pada algoritma Machine Learning apa pun. Fitur memberi peringkat berdasarkan skor statistik yang cenderung menentukan korelasi fitur dengan variabel hasil. Korelasi adalah istilah yang sangat kontekstual, dan bervariasi dari satu tugas ke tugas lainnya.

**2. Metode Wrapper**

Metode wrapper membutuhkan satu [jenis algoritma](https://www.trivusi.web.id/2022/06/jenis-jenis-algoritma.html) Machine Learning dan menggunakan kinerjanya sebagai kriteria evaluasi. Metode ini mencari fitur yang paling cocok untuk algoritma Machine Learning dan bertujuan untuk meningkatkan kinerja algoritma. Untuk mengevaluasi fitur, akurasi prediktif digunakan pada tugas klasifikasi.

Alur dari metode wrapper dapat dilihat pada gambar berikut:

|  |
| --- |
| [A diagram of a subscribed  Description automatically generated with medium confidence](https://www.analyticsvidhya.com/wp-content/uploads/2016/11/Wrapper_1.png) |
| *Sumber: analyticsvidhya.com* |

Metode wrapper didasarkan pada [algoritma](https://www.trivusi.web.id/2022/06/mengenal-algoritma.html) pencarian greedy karena metode ini mengevaluasi semua kemungkinan kombinasi fitur dan memilih kombinasi yang menghasilkan hasil terbaik. Kelemahan dari pendekatan ini adalah pengujian semua kemungkinan kombinasi fitur dapat menjadi sangat mahal secara komputasi, terutama jika himpunan fitur sangat besar.

Metode wrapper untuk pemilihan fitur dapat dibagi menjadi tiga kategori: **Forward selection**, **Backward elimination** dan **Recursive Feature elimination**

* **Forward Selection**: Metode seleksi berulang yang dimulai dengan fitur kosong pada model. Dalam setiap iterasi atau perulangan, kita menambahkan fitur yang memiliki pengaruh paling signifikan dalam meningkatkan model yang kita miliki. Kemudian dilanjutkan dengan penambahan variabel baru yang tidak meningkatkan kinerja model.
* **Backward Elimination**: Berkebalikan dengan metode forward selection, pada metode ini model berisi semua fitur. Kemudian pada setiap iterasi atau perulangan dilakukan penghapusan fitur yang tidak meningkatkan kinerja model secara signifikan. Kita mengulangi proses ini sampai model berisi fitur yang ideal, ditandai dengan tidak ada perubahan yang ditemukan ketika dilakukan penghapusan fitur.
* **Recursive Feature elimination**: Metode ini adalah optimasi algoritma greedy yang bertujuan untuk menemukan subset fitur berkinerja terbaik. Pada setiap iterasi, metode ini membangun model yang dimulai dari fitur paling kiri sampai semua fitur selesai dijelajahi. Metode ini mengabaikan fitur berkinerja terbaik atau terburuk di setiap iterasi. Sebaliknya metode ini memberi peringkat fitur berdasarkan urutan eliminasinya.

**3. Metode Embedded Selector**

Metode embedded selector adalah metode seleksi fitur yang menggabungkan keunggulan metode filter dan metode wrapper. Metode ini diimplementasikan oleh algoritma yang memiliki metode pemilihan fitur bawaannya sendiri.

|  |
| --- |
| [A black arrows pointing to a certain subject  Description automatically generated with medium confidence](https://www.analyticsvidhya.com/wp-content/uploads/2016/11/Embedded_1.png) |
| *Sumber: analyticsvidhya.com* |

Beberapa contoh paling populer dari metode ini adalah [regresi](https://www.trivusi.web.id/2022/07/jenis-analisis-regresi.html) Lasso dan ridge yang memiliki fungsi utama untuk mengurangi overfitting.

* **Regresi Lasso** melakukan regularisasi L1 yang menambahkan penalti ekuivalen dengan nilai absolut dari besaran koefisien.
* **Regresi Ridge** melakukan regularisasi L2 yang menambahkan penalti yang setara dengan kuadrat dari besarnya koefisien.