Customer Segmentation for Personalized Marketing in Food Delivery Services using Clustering and PCA

Problem Statement:

In the competitive food delivery industry, businesses often struggle to understand and engage their diverse user base effectively. Users differ widely in terms of age, spending habits, order frequency, app usage, and preferences. Without segmenting these users into meaningful groups, companies miss opportunities to personalize marketing, improve user satisfaction, and increase retention.

Project Goal:

The goal of this project is to **segment users of a food delivery app into meaningful clusters** based on their behavior and demographics using unsupervised learning techniques. By identifying distinct customer groups, the business can tailor marketing campaigns, improve service offerings, and optimize customer engagement strategies.

Learning Objectives

Data Understanding & Exploration

1. **Load and inspect** food delivery user data to understand variable types (e.g., numeric vs. categorical), ranges, and distributions.
2. **Summarize and interpret** key statistical properties (mean, median, standard deviation, skewness) of user behavior features like Age, Total Orders, Average Spend, etc.
3. **Visualize** data distributions using histograms, box plots, and pair plots to explore potential groupings or outliers.
4. **Identify relationships** between user variables (e.g., App Usage Time and Average Spend) through correlation matrices and scatter plots.

Data Cleaning & Preprocessing

5. **Drop or encode** irrelevant or categorical features (e.g., UserID, FavoriteCuisine) as appropriate for unsupervised learning models.
6. **Standardize** numerical features using StandardScaler to ensure equal contribution to distance-based clustering.
7. **Evaluate the effect** of scaling and dimensionality reduction on the data’s structure through pre- and post-processing visualizations.

Dimensionality Reduction with PCA

8. **Apply Principal Component Analysis (PCA)** to reduce high-dimensional data to 2D for effective visualization of user clusters.
9. **Interpret principal components** by examining feature loadings to understand which user behaviors contribute most to variance.
10. **Visualize** PCA-transformed data to observe potential natural clusters and separation of user types.

Unsupervised Learning: Clustering

11. **Implement K-Means clustering** on scaled user data and interpret resulting clusters in the context of user behavior.
12. **Determine the optimal number of clusters** using the Elbow Method or Silhouette Score.
13. **Apply Agglomerative (Hierarchical) Clustering** and **visualize** results using dendrograms to understand hierarchical relationships among users.
14. **Compare clustering results** from K-Means and Hierarchical Clustering to identify consistency or divergence in groupings.

Model Evaluation & Visualization

15. **Label PCA scatter plots** with cluster assignments to visually interpret how users are grouped in reduced dimensions.
16. **Analyze cluster characteristics** by computing average values of features within each cluster (e.g., high spenders, frequent users).
17. **Create a cluster profile summary** to describe each segment using business-relevant insights (e.g., “young frequent users,” “older high spenders”).

Critical Thinking & Real-world Framing

18. **Frame the clustering output** as a customer segmentation strategy for personalized marketing or product recommendations.
19. **Explain how unsupervised learning** helps businesses make data-driven decisions in the absence of labeled training data.
20. **Reflect on limitations** of clustering models (e.g., sensitivity to scaling, shape of clusters) and suggest ways to improve (e.g., DBSCAN, inclusion of more behavioral data).

Dataset

https://samatrix-data.s3.ap-south-1.amazonaws.com/ML/food_delivery.csv

What to Submit:

Submission Type: Individual

Each student must submit the following:

1. **Jupyter Notebook (.ipynb file) or python filr (.py file)**
   a. Filename: YourFullName_FoodDelivery.ipynb (e.g., AnanyaKumar_ FoodDelivery.ipynb)
   b. Your notebook must follow the steps and structure discussed in class following the instructions in the submission guideline
2. **Word or PDF file**
   a. Answers “Questions for Report” in separate file

Step 1 – Importing Libraries

Tool / Library	What It Helps You Do
pandas	Load and view your dataset like a spreadsheet
StandardScaler	Prepare your data so numbers are on the same scale

PCA (Principal Component Analysis)	Shrink your dataset to just 2 main features so you can visualize it
KMeans, AgglomerativeClustering, DBSCAN	Try different ways to group similar users together
dendrogram, linkage	Show a tree of how users are grouped in a hierarchy
matplotlib, seaborn	Create beautiful graphs to understand your results

Step 2 – Load Data

In this part of the project, you will **open the dataset**, **look at the first few rows**, and **understand what kind of data you are working with**. Think of it like opening a file for the first time and scanning through the first page to get a sense of what's inside.

You are working with a file called **food_delivery.csv**. This file contains data about customers using a food delivery app.

Each **row** in the file represents **one customer**, and each **column** describes some detail about that customer — for example:

- How many times they've ordered
- How much they usually spend
- What kind of food they like

This file is saved in a common format called **CSV (Comma-Separated Values)**. It's like an Excel sheet where each row is a customer and each column is a piece of information.

Step-by-Step Instructions

Step 1: Load the File into Your Data Tool

- Open your data analysis tool (like Jupyter Notebook or Google Colab).
- Use the tool's option to read or upload the food_delivery.csv file.
- Once loaded, the dataset will be stored in a table format called a **DataFrame** — think of it like a smart version of an Excel table.

Step 2: View the First 5 Rows

- Use your tool to display the first 5 rows of the dataset.
- This is called a "data preview."
- You'll see a table with customer details, and it will help you understand:
  - What kind of information is available
  - Whether the data loaded correctly
  - What the column names are (e.g., UserID, TotalOrders, AvgSpend, etc.)

Questions for Report

1 What kind of information is available in the dataset?
List a few columns from the dataset and explain in your own words what each one tells us about the user.

2 How many rows and columns does the dataset have?
What does this tell you about the size of the dataset and how much information is available for analysis?

3 Can you spot any unusual values or missing information?
If yes, explain which column and what type of issue you noticed.

4 Choose any one user (a row) and describe their behavior.
What can you tell about this user based on the numbers or categories in their row?

5 What patterns or questions come to your mind after seeing the first few rows?
For example: Do some users spend more? Do some users order more often?

Step 3 –Summary Statistics

In this step, you'll take a **quick overview** of your entire dataset by generating **summary statistics**. This helps you understand the **shape, spread, and patterns** in your data before moving on to clustering or modeling.

Step-by-Step Instructions

Step 1: Generate the Summary Statistics

- Use your data tool (like Jupyter or Colab) to **summarize all numeric columns** in the dataset.
- You will get statistics like:
    - **Count** – how many users have values in that column
    - **Mean** – the average value (e.g., average age, average spend)
    - **Standard Deviation** – how much the values vary from the average
    - **Minimum and Maximum** – the smallest and largest value
    - **Percentiles (25%, 50%, 75%)** – tells you how values are spread out (like what's "typical")

Questions for Report

1 What is the average (mean) value of two or more key columns?
Example: What is the average age of users? What is the average amount they spend per order?
2 Which column has the highest variation (difference between users)?
Look at the standard deviation (std) values — which feature varies the most among users, and what might that mean?

3 What are the minimum and maximum values in the dataset, and what do they tell you?
Are there users who spend very little or a lot? Who gives the lowest or highest delivery ratings?

4 Pick any one column and describe how user behavior is spread out.

Look at the 25%, 50%, and 75% values (called percentiles or quartiles) and explain what that tells you about most users.

5 Based on the summary statistics, what are 2-3 possible customer groups or behaviors you think might exist?
For example: Are there heavy spenders and light spenders? Are there frequent users and casual users?

Step 4: Check for Missing Values in Your Data

In this step, you will **check your dataset for missing values** — empty or blank cells where important information may be missing. This is a crucial step before doing any serious data analysis or clustering.

Imagine you're organizing a school event and you have a list of participants. If someone forgot to fill in their name or dietary preferences, it might lead to confusion or mistakes. The same happens in data analysis:

- Missing values can lead to **incorrect results**
- Some machine learning models might even **fail to run**
- You might make **wrong assumptions** about your users

So, before doing anything else — you need to ask:
"Do I have all the information I need for every user?"

Step-by-Step
Step 1: Run a Check for Missing Data

- You will run a simple command that **scans every column** in your dataset.
- It will **count how many values are missing** (if any) in each column.

Questions for Report

1 Were there any missing values in your dataset?
If yes, list the columns where data was missing and how many values were missing in each.

2 Why is it important to check for missing values before analyzing data?
Explain in your own words why having complete or incomplete data matters in a project.

3 If you found missing values, what would you do about them?
Would you remove those rows? Fill in with an average? Ignore them? Explain your choice.

4 What could be some real-life reasons why data might be missing in a food delivery app?
For example: Why might a user's favorite cuisine or delivery rating be blank?

5 How does knowing your data has no missing values help your project?
Explain how complete data benefits the next steps like clustering or visualization.

Step 5: Keep Only Useful Columns

In this step, you will **remove columns that are not useful for analysis** and keep only the important ones that will help you group users based on similar behavior. This prepares your data for the next steps like scaling and clustering.

Imagine you're organizing students into groups for a class project. Would you group them based on:

- their name tags (like "Student #42")?
- or their actual skills and interests?

Of course, you'd use real information — not random ID numbers.
Similarly, in this project, you want to cluster users based on:

- how often they order,
- how much they spend,
- how long they use the app,
- and how happy they are with deliveries.

You do **not** want to use columns like:

- **User ID** (just a unique number)
- **Favorite Cuisine** (text, not a number — and we're not analyzing it right now)

1: Review Your Dataset Columns

- Take a moment to **look at all the columns** in your dataset.
- Ask yourself: "Which columns describe user behavior in numbers?"
- You're likely to see columns like:
  - Age
  - Total Orders
  - Average Spend
  - Delivery Rating
  - App Usage Time

2: Identify Columns to Remove

- You should **remove** columns that are:
  - Just labels (e.g., UserID)
  - Text-based (e.g., FavoriteCuisine) — because clustering methods work best with **numbers**

3: Create a Clean Version of the Data

- After removing the unhelpful columns, your dataset is now clean and ready for the next steps.
- Save this cleaned version separately (often given a name like X) so that:
  - You don't affect the original dataset
  - You use only the **useful numeric data** for further analysis

Questions for Report

1 Which columns did you remove from the dataset, and why?
Explain in simple words why those columns were not useful for clustering or analysis.

2 Which columns did you keep in your cleaned dataset?
List 3 or more columns and explain what each one tells you about user behavior.

3 Why should unique IDs (like UserID) not be used for clustering?
Explain what happens if you try to group people based on IDs that are just labels.

4 Why is it helpful to remove text columns (like FavoriteCuisine) in early clustering steps?
What challenges might text data cause when using machine learning models that expect numbers?

5 How does this data-cleaning step help you get ready for clustering users?
In your own words, explain why having clean, number-based data is important before applying clustering techniques.

Step 6: Standardize Your Data

In this step, you will **standardize (or scale)** your data so that all the features (like age, total orders, app usage, etc.) are on the **same scale**. This ensures that no single column unfairly influences your clustering results just because it has bigger numbers.

Imagine you're forming student groups for a workshop. You want to group students based on:

- Their **age**
- Their **grades**
- And their **hours of volunteering**

Now, suppose:

- Age ranges from **18 to 60**
- Grades range from **0 to 100**
- Volunteering hours range from **0 to 500**

If you group them directly based on these numbers, **volunteering hours will dominate**, just because the numbers are bigger — **not necessarily because they're more important**.
This is why we **scale** all the values so they're **fair and equal** before we group people.

1: Understand What Standardization Means

- Standardization means **transforming all values** so that:
    - The **average becomes 0**
    - The **spread of the values becomes 1**
- It keeps the shape of the data but puts all columns on **the same scale**.

2: Apply Standardization to the Cleaned Data

- You will use a tool to **look at each column**, learn the average and variation, and then **adjust all the numbers** accordingly.
- After this, every feature (like age, spend, orders) will have numbers around 0 — with negative numbers for below-average values and positive numbers for above-average values.

3: Save This New Scaled Data for Later Use

- This newly standardized version of your data will be saved into a new table.
- This is what you will use for clustering in upcoming steps.

Questions for Report

1 Why do we need to standardize (scale) the data before applying clustering?
Explain in your own words why it's important to make all features use the same scale.

2 What could happen if we skip the scaling step and apply clustering directly?
Give an example using two features like "App Usage Time" and "Delivery Rating" to explain your answer.

3 Which columns in your dataset do you think had the largest numbers?
And which ones had smaller numbers? Why is that a problem for clustering if not scaled?

4 After scaling, what changes in your data? What stays the same?
Hint: Are the numbers different? Are the patterns or relationships lost?

5 How does standardization help you get more accurate and fair clusters?
Write a short reflection on how this step prepares your dataset for grouping similar users.

Step 7: Reduce Dimensions Using PCA

In this step, you will simplify your dataset by reducing it from many columns (or features) to just **two main features** using a method called **Principal Component Analysis (PCA)**.
This makes your data:

- Easier to **visualize**
- Faster to **process**
- And still rich with useful **patterns**

Imagine you're trying to describe every customer using 5 or 6 details like:

- Age
- Total orders
- Average spend
- Delivery rating
- App usage time

That's a lot to look at all at once. PCA helps by saying:
"Let's find **just two powerful features** that summarize most of this information."
These new features are combinations of the original ones but still reflect the overall behavior of each customer.

Step by Step Instructions

1: Choose How Many Features to Keep

- Decide to keep just **2 new features** (called **principal components**).

- These components will represent most of the variation (differences) in the user data.

2: Let PCA Analyze Your Scaled Data

- PCA looks at your already scaled data and finds **which directions (or combinations of features)** explain the most.
- For example, PCA might combine "App usage time" and "Average spend" into a new feature that represents "User engagement level".

3: Replace the Original Data with the Simplified Version

- After PCA is done, each user will now have **2 new values** (instead of 5 or 6).
- These values are easier to **visualize in a 2D chart** and still capture **most of the behavior patterns**.

Example

Imagine you're describing cities. Instead of listing:

- Population
- Number of cars
- Pollution level
- Traffic speed

You summarize it in just 2 terms:

1. Urban intensity
2. Transport efficiency

That's what PCA does — it simplifies **complex things into fewer, powerful insights**.

Questions for Report

1 What is the main purpose of using PCA in this project?
Explain in your own words why we reduced the dataset to 2 features.

2 How many features (columns) did your data have before PCA? And how many after?
Why do you think it's helpful to reduce the number of features before clustering?

3 Do you think any important information might be lost during PCA? Why or why not?
Reflect on the trade-off between simplification and completeness.

4 What benefits does PCA provide when you want to visualize your user data?
How does reducing the data to two dimensions help us "see" patterns more clearly?

5 Imagine you had to explain PCA to a friend. How would you describe what it does in one or two sentences?
Try to use a simple real-life example (like summarizing school subjects or city data).

Step 8: Group Users Using KMeans Clustering

In this step, you will use a technique called **KMeans Clustering** to group your app users into **3 clusters (or user segments)** based on their behaviors like spending, app usage, order history, and ratings.
This helps answer questions like:
"Are there different types of users in our dataset — like budget buyers, frequent users, or loyal customers?"

What Is Clustering?
Clustering is a way of **grouping people or things based on how similar they are**.
Real-world example:
Think of organizing books in a library:

- Fiction in one section
- Biographies in another
- Science books in a third

You don't label every book yourself — you **look at the content** and decide what group it fits into.
KMeans does something similar with your users:
It **analyzes their behavior and automatically groups them** into categories — without you having to label anything.

Step by Step Instructions
1: Decide How Many Groups You Want

- You will ask the tool to **create 3 groups (clusters)**.
- This is like saying: "I want to divide users into 3 customer types based on their behavior."

2: Let the Tool Analyze and Group the Users

- The tool will:
  - Look at the **scaled data** you prepared earlier
  - Measure how similar or different users are from one another
  - Group them based on those patterns

Every user will now be assigned to **one of the 3 groups**.

3: Save the Group Assignments

- Each user now has a label (like Group 0, Group 1, or Group 2).
- These labels tell you **which group each user belongs to**.
- You will use these labels later when plotting and analyzing the clusters.

Questions for Report

1 What is the main purpose of using KMeans clustering in this project?
In your own words, explain why we grouped users and what KMeans helps us discover.

2 How many clusters did you create, and why?
Explain why you chose that number of user groups. Do you think it was a good choice?

3 What does each cluster label represent?
For example, what kind of users might be in Cluster 0, Cluster 1, or Cluster 2?

4 How do you think this grouping could help a business?
Give one or two ideas on how these user clusters could be used to improve marketing, service, or app features.

5 Were you surprised by anything after applying clustering? Why or why not?
Reflect on whether the groups made sense to you, and what you might want to explore further.

Step 9: Visualize the Clusters Using a Scatter Plot

In this step, you will **draw a 2D scatter plot** to **see the user groups (clusters)** formed by the KMeans algorithm. This plot helps you **visually understand how your app users are grouped**, and how different or similar those groups are.

Step by Step Instructions

1: Set Up the Plot Area

- Think of this as choosing the size of the canvas where your dots (users) will be placed.
- You'll decide how wide and tall your plot should be so it's easy to read.

2: Plot the Users on the Chart

- Each user will appear as a **dot** based on their two main behavior features (PC1 and PC2).
- These features are created from the PCA step and represent the **overall behavior patterns** of the users.

3: Color the Dots Based on Clusters

- You will use the KMeans results to **color-code** each user:
    - Cluster 0 (e.g. casual users)
    - Cluster 1 (e.g. loyal users)
    - Cluster 2 (e.g. low-engagement users)
- This makes it easy to **see which users are similar** and where each group is located.

4: Add Labels and a Title

- Add a title to your chart so others understand what it shows.
- Label your x-axis and y-axis with "PC1" and "PC2", which represent the behavior features from PCA.

5: Display the Chart

- Once everything is ready, you'll display the chart.
- You should now **see a colorful scatter plot** where users are grouped based on how they behave on the app.

What You Should Expect to See

- A chart with **three distinct colors**, each representing a different user group.
- Clusters (groups) that are **close together** suggest similar users.
- If clusters are **far apart**, it means the behavior differences are bigger.

Questions for Report

1 What does each dot in your scatter plot represent?
Explain what each point on the chart stands for and why it's plotted in that specific position.

2 How many clusters do you see in the chart, and how are they separated?
Are the clusters clearly apart from each other, or are some overlapping? What might this tell you?

3 Choose one cluster and describe what kind of users might belong to it.
Based on where the cluster is located, what do you think those users do differently?

4 Why is it helpful to visualize clusters like this instead of just looking at raw numbers?
What does the chart help you see or understand that tables and statistics might not?

5 If you were running a food delivery app, how could this chart help you improve your service?
Give one practical idea based on what you learned from the cluster plot.

Step 10: Group Users with Agglomerative Clustering

In this step, you will explore a new way to group your users — called **Agglomerative Clustering**. This technique helps you understand how users naturally form **hierarchies** or **step-by-step clusters** based on similar behaviors.
Think of it as:
“Let’s start with every user in their own tiny group, and then slowly merge the most similar ones together until we end up with 3 big clusters.”

Until now, you used **KMeans** to group users based on behavior. That method finds clusters by jumping straight into dividing users. In this step, you’ll take a **different approach**: you’ll **build the groups gradually**, step by step — like fitting puzzle pieces together.
This helps you:

- See how users naturally combine into small and large groups
- Compare the results of two clustering methods
- Understand which clustering method gives more meaningful insights

Step by Step Instructions

1: Decide the Number of Clusters

- You will tell the algorithm to create **3 clusters**.

- This means you expect the users to fall into **three main types or groups** based on their behavior.

2: Apply Agglomerative Clustering

- The algorithm starts with **every user in their own separate group**.
- It looks for the **most similar users** and **joins them together** step by step.
- This continues until only 3 big user groups remain.

3: Assign Group Labels to Each User

- After the merging is complete, each user will be **assigned a group number** (like 0, 1, or 2).
- This label tells you which group that user belongs to.

4: Save These Labels for Analysis

- These group labels are important because:
  - You'll use them in your visualizations
  - You can describe how each group behaves
  - You can compare them with the clusters formed by KMeans

Example

Imagine you're sorting books:

- You start by putting each book on its own shelf
- Then you slowly group similar ones — all history books, all science books, all novels
- Finally, you have 3 main categories

That's how Agglomerative Clustering works — **it builds clusters gradually**.

Questions for Report

1 What is Agglomerative Clustering and how does it work?
Explain in your own words how this method forms user groups step by step.

2 How is Agglomerative Clustering different from KMeans Clustering?
Compare how both methods create clusters. What did you notice when switching from KMeans to this method?

3 How many clusters did you create, and how did the user distribution look?
Were the groups balanced or was one group much larger or smaller? What might this mean?

4 Choose one cluster and describe the possible behavior of users in that group.
Based on your knowledge of the dataset, what kind of users might be grouped together here?

5 Which clustering method (KMeans or Agglomerative) do you feel gave better groupings — and why?
Share your opinion and support it with one reason based on what you saw in the project.

Step 11: Visualize Clusters Using Agglomerative Clustering

In this step, you will create a **scatter plot** that shows how users are grouped based on the **Agglomerative Clustering algorithm**. This helps you understand user behavior visually and compare it with the clusters you saw earlier using KMeans.

Earlier, you grouped users into clusters using a method called **Agglomerative Clustering** — a technique that builds groups by **slowly merging similar users** together.
Now, it's time to **visualize those groupings** in a 2D chart to:

- See how users are grouped together
- Identify patterns in user behavior
- Compare the results with KMeans clustering
- Prepare for interpretation and business insights

Step by Step Instructions

1: Set Up the Plot

- Think of this like choosing the size of the paper or whiteboard where you'll draw your scatter plot.
- You'll make it wide enough to clearly show all the users.

2: Plot Each User as a Dot

- Each dot represents one user in your dataset.
- The **horizontal axis (PC1)** and **vertical axis (PC2)** represent the main behavior patterns, created earlier using PCA.
- These axes help you simplify and visualize multi-feature data in just two dimensions.

3: Add Color to Show Clusters

- You'll color each dot based on the **cluster** the user belongs to, using the results from Agglomerative Clustering.
- For example:
  - Blue = Cluster 0
  - Gray = Cluster 1
  - Red = Cluster 2
- The colors make it easy to **see which users belong together**.

4: Label the Axes and Add a Title

- Add a title to your chart so it's clear what you're showing (e.g., "PCA with Agglomerative Clustering").
- Label the X-axis as "PC1" and the Y-axis as "PC2" — these are the simplified dimensions of user behavior.

5: Display the Chart

- Once everything is ready, you'll display the chart.
- Now you can **see the clusters visually** and begin interpreting them.

Example

Imagine plotting students on a chart where:

- X-axis = how often they study
- Y-axis = how much time they spend relaxing

Now, if you color them by **study group**, you'll easily spot:

- Studious students
- Balanced ones
- More relaxed ones

This is exactly what you're doing — but with **food delivery users**.

Questions for Report

1 What does each dot in the scatter plot represent?
Explain what one dot means in the context of this project and why it's important.

2 How many clusters are shown in the plot? Do they appear clearly separated or overlapping?
Describe the overall shape and clarity of the clusters.

3 Pick one cluster and describe what kind of users it might represent.
Use the position and spread of the dots to guess the behavior pattern of that group.

4 How does this clustering result compare to what you saw with KMeans?
Were the clusters similar or different? Which one felt more meaningful to you, and why?

5 Why is visualizing user clusters important for a company like a food delivery app?
Think about how this chart could help in decision-making, marketing, or service improvement.

Step 12: Build a Dendrogram to Visualize Cluster Formation

In this step, you'll **draw a tree-shaped chart** (called a **dendrogram**) to see **how users were grouped together** using hierarchical clustering. This tree helps you figure out how similar different users are, and how many meaningful clusters you should keep.

- A dendrogram gives you a **zoomed-out view** of all the clustering decisions.
- It shows you **how and when users merged into groups**, from start to finish.
- You can visually decide **how many clusters** best represent your data.

Think of it like watching how a **family tree** forms, step by step.

Step by Step Instructions:

1: Prepare the Tree Data

- First, you will use a clustering method to prepare **how the tree should be built**.
- The computer checks how **similar or different each user is** from others based on their behavior (like app usage, spending, delivery rating, etc.).
- Users who are most alike are grouped first. Then groups merge into bigger groups.

2: Set the Chart Size

- You'll create a wide enough space (like a digital whiteboard) to display your tree clearly.
- This helps you see how small and big clusters form.

3: Draw the Dendrogram (Tree)

- Now you will plot the tree:
  - Each **bottom point** is a user.
  - As you go up, similar users or groups are **connected together**.
- You'll only show the **last 30 groups** to keep the chart readable.

4: Add Labels and Show the Tree

- Add a title and axis labels to explain what your chart shows.
- The **Y-axis** shows how different users or groups are when they merged.
- Finally, display the chart and observe the patterns.

Example:

Think of arranging books on a shelf:

1. First, group books with the exact same topic (like all history books).
2. Then group similar genres (like all non-fiction).
3. Finally, you place all books into 3 or 4 broad categories.

A dendrogram works the same way — it **builds clusters step-by-step**, and the tree helps you decide **how many final groups to keep**.

What You Should Look For:

- Where do the **big jumps in height** happen in the tree? That's a clue to **where to "cut" the tree** to form final clusters.
- Do some users or groups merge at **very low height**? That means they are **very similar**.
- Are there **5 or fewer big branches** at the top? That might mean **you have around 5 natural clusters**.

Questions for Report

1 What is a dendrogram and what does it show?

Explain in your own words what a dendrogram is and why it is used in clustering.

2 How many user groups (clusters) do you think are visible in the dendrogram?

Based on your observation, where would you draw a horizontal line to divide the tree into clear groups?

3 What does it mean when two users or groups are merged at a very low height?

Give an example from the dendrogram and explain what it tells you about user similarity.

4 Was the dendrogram result similar or different compared to KMeans and Agglomerative clustering visual plots?

Briefly compare and say which method gave clearer groups and why you think so.

5 How could a business (like a food delivery company) use these cluster insights in real life?
Suggest one way this clustering information could help improve marketing, app features, or user satisfaction.

Step 13: Discover User Patterns with DBSCAN Clustering

In this step, you'll use a powerful clustering method called **DBSCAN** to discover **natural groups** in your user data — and even **spot users who don't fit in any group** (outliers). Unlike previous methods where you had to choose the number of clusters, **DBSCAN figures it out automatically** based on how close users are to one another.

- DBSCAN helps you find **organic user behavior patterns** — especially when groups are irregular in shape or size.
- It also detects **outliers**, which are users whose behavior is very different from others.
- Businesses can use this to **find special cases** — such as very active users, rare shoppers, or even suspicious activity.

Step by Step Instructions:

Step 1: Choose the Right Settings for DBSCAN
You need to help the DBSCAN model understand:

- **How close users need to be to be grouped together** (called eps)
- **How many users must be close together to form a group** (called min_samples)

Think of it like this:

- You're looking for groups of users sitting near each other in a classroom.
- If 5 or more students are sitting close enough, you consider them a group.
- Anyone sitting too far from everyone is not part of any group — they are **outliers**.

Step 2: Apply DBSCAN to Your User Data
You'll now run the clustering algorithm on the scaled version of your dataset (where all the numbers have already been adjusted to the same range). The DBSCAN method will:

- Scan the data point-by-point
- Form clusters where enough users are close together
- Label users with a group number (like Cluster 0, Cluster 1…)
- Label outliers with **-1** (means "not part of any cluster")

Step 3: Capture the Cluster Labels
The result will be a list showing **which user belongs to which group** — or if the user is an **outlier**.
Here's how it might look in simple terms:

- User 1 → Cluster 0
- User 2 → Cluster 0
- User 3 → Cluster 1
- User 4 → Not in any group (outlier)

Step 4: Prepare for Visualization

After DBSCAN is done grouping the users, the next step will be to **visualize the results**. You'll make a scatter plot to show:

- Which users belong to each cluster (by color)
- Which ones are outliers (usually shown in a separate color)

Example:

Imagine a food delivery app is analyzing users:

- DBSCAN might find a group of users who order every weekend (Cluster 0)
- Another group might order only on weekdays (Cluster 1)
- But some users order at random times or only once — DBSCAN marks them as **outliers**

These outliers might need **special attention**, **marketing**, or even **fraud checks** depending on the context.

Questions for Report

1 What is DBSCAN, and how is it different from KMeans or Agglomerative Clustering?

Write in your own words how DBSCAN forms groups and how it handles unusual users.

2 How many user groups (clusters) were formed by DBSCAN?

List the number of clusters you found, and how many users were in each.

3 Did DBSCAN identify any outliers? If yes, how many and what might these users represent?

Think about what kind of users might not fit into any group and why.

4 Compare the DBSCAN results with your earlier clustering (KMeans or Hierarchical). Were the groupings similar or different?

Which method gave you clearer or more useful clusters in your opinion?

5 How could a company use the DBSCAN results to improve its food delivery service or marketing?

Suggest one way the insights from these clusters or outliers could help in decision-making.

Step 14: Visualize DBSCAN Clustering with a Scatter Plot

In this step, you'll **create a colorful scatter plot** to show how users were grouped using the **DBSCAN clustering method**. This helps you easily see which users fall into groups and which ones are outliers (users who don't fit in any group).

- You've already applied DBSCAN to group users based on behavior (like spending, app usage, etc.).
- But numbers alone don't tell the full story — **seeing the clusters visually helps you understand the patterns better**.
- You can also **spot outliers instantly** — users who act very differently.

Step by Step Instructions:

1: Prepare the Plotting Space
You'll first set up a "canvas" or drawing space large enough to show your users clearly.

2: Plot the Users on a 2D Graph

- Each **dot** on the graph will represent one user.
- The position of the dot comes from the **PCA transformation**, which turns complex user data into two simple dimensions:
    - **PC1 (X-axis)** = principal component 1
    - **PC2 (Y-axis)** = principal component 2

This way, you can **visualize all users in 2D**, even if the original data had many features.

3: Color the Users by Their Group (Cluster)

- You'll color each user according to the **DBSCAN group** they belong to.
- **Same color = same group.**
- Outliers (users who don't belong to any cluster) will be shown in a different color.

4: Add Labels and Show the Plot

- Add a title to explain the chart ("PCA with DBSCAN Clustering").
- Label the axes to show what PC1 and PC2 mean.
- Add a **legend** to explain what each color represents.
- Finally, show the plot so you can interpret it.

Example:
Imagine organizing a big party:

- People who chat with each other form groups (clusters).
- Someone sitting alone in the corner? That's an outlier!
- Your scatter plot is like a **seating map** that shows who's part of which friend group — and who's sitting alone.

Questions for Report

1 How many distinct groups (clusters) can you see in the scatter plot?
Look at the colors on the plot. How many clusters were formed?

2 Are there any outliers (users marked as -1)? Where do they appear in the plot?
Describe where the outliers are placed. Are they close to other users or far away?

3 Does the shape or size of the cluster suggest anything about user behavior?
For example: Are most users tightly packed? Spread out? What does that tell you?

4 How does this DBSCAN visualization compare to the KMeans or Agglomerative Clustering plot?
Do the clusters look similar or different? Which one gives clearer groups?

5 If you were working in the marketing team of a food delivery app, how would you use this visual insight to improve service or engagement?

Think about what this user grouping can tell you about customer needs.