

## Problem Statement: Efficiently Managing Public Event Feedback in a Lodha Belair

### Context:

In a multiplex society, people come from diverse cultural, social, and economic backgrounds, which often leads to differing preferences, tastes, and feedback regarding events, entertainment, and activities. Organizing public events such as community gatherings, theater performances, or festivals requires the collection and analysis of feedback from attendees. However, in a multiplex society, feedback can be highly varied and challenging to manage due to the diverse demographics.

### Problem:

You are tasked with creating a Python program that can help organizers efficiently collect and categorize feedback from a diverse group of event attendees. The goal is to collect basic event feedback (e.g., rating the event, providing comments) from attendees, and then categorize it based on the attendee's demographic background (e.g., age group, gender, occupation, etc.).

The challenge is to **efficiently** group feedback by demographic categories (like age groups, gender, occupation), analyze it, and generate simple reports that show trends in satisfaction, areas of improvement, and general sentiment.

### Input:

1. A list of feedback responses from attendees. Each response contains:
  - **Rating:** Rating of the event (e.g., 1-5, where 1 is "Very Poor" and 5 is "Excellent").
  - **Comments:** A short textual comment about the event.
  - **Age group:** The age group of the attendee (e.g., "18-25", "26-35", "36-50", "50+").
  - **Gender:** The gender of the attendee (e.g., "Male", "Female", "Other").
  - **Occupation:** The occupation of the attendee (e.g., "Student", "Working Professional", "Retired").

Example data:

```
feedback_data = [  
    {"rating": 5, "comments": "Amazing performance! Loved every minute of it.", "age_group":  
"18-25", "gender": "Male", "occupation": "Student"},  
    {"rating": 3, "comments": "It was okay, could be better.", "age_group": "26-35", "gender":  
"Female", "occupation": "Working Professional"},  
    {"rating": 4, "comments": "Great event, but the seating could have been better.",  
"age_group": "36-50", "gender": "Male", "occupation": "Working Professional"},  
    {"rating": 2, "comments": "Not worth the money. Very disappointing.", "age_group": "50+",  
"gender": "Female", "occupation": "Retired"},
```

```
{ "rating": 5, "comments": "Fantastic! Will definitely come again!", "age_group": "18-25",  
  "gender": "Female", "occupation": "Student"},  
{ "rating": 1, "comments": "Terrible experience, never attending again.", "age_group":  
  "26-35", "gender": "Male", "occupation": "Working Professional"},  
]
```

### Task:

1. **Categorize feedback** by **age group**, **gender**, and **occupation**.
2. **Calculate the average rating** for each demographic group (e.g., average rating for "18-25, Male, Student").
3. **Identify trends** in the comments by counting the occurrence of positive or negative words (e.g., "great", "terrible", "disappointing", "fantastic").
4. **Generate a simple report** that summarizes:
  - The average rating for each demographic group.
  - The most common positive and negative feedback keywords across groups.
  - How the different demographics felt about specific aspects of the event (e.g., "seating", "performance quality", "value for money").

### Output:

The output should be a summary that looks something like this:

--- Feedback Summary ---

Average Rating by Demographic Group:

- Age Group 18-25, Gender Male, Occupation Student: 5.0
- Age Group 26-35, Gender Female, Occupation Working Professional: 3.0
- Age Group 36-50, Gender Male, Occupation Working Professional: 4.0
- Age Group 50+, Gender Female, Occupation Retired: 2.0
- Age Group 18-25, Gender Female, Occupation Student: 5.0
- Age Group 26-35, Gender Male, Occupation Working Professional: 1.0

Most Common Positive Words:

- Amazing: 1
- Fantastic: 2
- Great: 2
- Loved: 1
- Excellent: 1

Most Common Negative Words:

- Terrible: 1
- Disappointing: 1
- Could be better: 1

Overall Sentiment:

- Positive Sentiment: 4 responses
- Neutral Sentiment: 0 responses

- Negative Sentiment: 2 responses

---

## Solution Outline Using Basic Python:

```
# Sample feedback data
feedback_data = [
    {"rating": 5, "comments": "Amazing performance! Loved every minute of it.", "age_group":
"18-25", "gender": "Male", "occupation": "Student"},
    {"rating": 3, "comments": "It was okay, could be better.", "age_group": "26-35", "gender":
"Female", "occupation": "Working Professional"},
    {"rating": 4, "comments": "Great event, but the seating could have been better.",
"age_group": "36-50", "gender": "Male", "occupation": "Working Professional"},
    {"rating": 2, "comments": "Not worth the money. Very disappointing.", "age_group": "50+",
"gender": "Female", "occupation": "Retired"},
    {"rating": 5, "comments": "Fantastic! Will definitely come again!", "age_group": "18-25",
"gender": "Female", "occupation": "Student"},
    {"rating": 1, "comments": "Terrible experience, never attending again.", "age_group":
"26-35", "gender": "Male", "occupation": "Working Professional"},
]
```

```
# Function to calculate average rating for each demographic
def calculate_average_rating(feedback_data):
    demographics = {}

    for feedback in feedback_data:
        key = (feedback["age_group"], feedback["gender"], feedback["occupation"])
        if key not in demographics:
            demographics[key] = {"ratings": [], "comments": []}
        demographics[key]["ratings"].append(feedback["rating"])
        demographics[key]["comments"].append(feedback["comments"])

    averages = {}
    for key, data in demographics.items():
        avg_rating = sum(data["ratings"]) / len(data["ratings"])
        averages[key] = avg_rating

    return averages
```

```
# Function to identify most common positive and negative words in comments
def identify_feedback_keywords(feedback_data):
    positive_keywords = ["amazing", "fantastic", "great", "loved", "excellent", "awesome"]
    negative_keywords = ["terrible", "disappointing", "could be better", "worst", "awful"]

    positive_count = {word: 0 for word in positive_keywords}
    negative_count = {word: 0 for word in negative_keywords}
```

```

for feedback in feedback_data:
    comment = feedback["comments"].lower()
    for word in positive_keywords:
        if word in comment:
            positive_count[word] += 1
    for word in negative_keywords:
        if word in comment:
            negative_count[word] += 1

return positive_count, negative_count

# Calculate average ratings
averages = calculate_average_rating(feedback_data)

# Identify feedback keywords
positive_keywords, negative_keywords = identify_feedback_keywords(feedback_data)

# Print results
print("--- Feedback Summary ---")
print("Average Rating by Demographic Group:")
for demographic, avg_rating in averages.items():
    print(f"- Age Group {demographic[0]}, Gender {demographic[1]}, Occupation {demographic[2]}: {avg_rating:.1f}")

print("\nMost Common Positive Words:")
for word, count in positive_keywords.items():
    print(f"- {word.capitalize()}: {count}")

print("\nMost Common Negative Words:")
for word, count in negative_keywords.items():
    print(f"- {word.capitalize()}: {count}")

# Simple sentiment analysis based on average rating
positive_count = sum(1 for rating in [feedback["rating"] for feedback in feedback_data] if rating >= 4)
negative_count = sum(1 for rating in [feedback["rating"] for feedback in feedback_data] if rating <= 2)

print(f"\nOverall Sentiment:")
print(f"- Positive Sentiment: {positive_count} responses")
print(f"- Negative Sentiment: {negative_count} responses")

```

---

## Expected Output:

```

--- Feedback Summary ---
Average Rating by Demographic Group:

```

- Age Group 18-25, Gender Male, Occupation Student: 5.0
- Age Group 26-35, Gender Female, Occupation Working Professional: 3.0
- Age Group 36-50, Gender Male, Occupation Working Professional: 4.0
- Age Group 50+, Gender Female, Occupation Retired: 2.0
- Age Group 18-25, Gender Female, Occupation Student: 5.0
- Age Group 26-35, Gender Male, Occupation Working Professional: 1.0

Most Common Positive Words:

- Amazing: 1
- Fantastic: 1
- Great: 1
- Loved: 1
- Excellent: 1
- Awesome: 0

Most Common Negative Words:

- Terrible:

Here is a breakdown of the data types used in the solution:

## 1. **feedback\_data** (List of Dictionaries)

- **Type:** `list`
- **Description:** A list that contains dictionaries, where each dictionary represents a feedback response from an individual attendee.
- **Example:**

```
feedback_data = [
    {"rating": 5, "comments": "Amazing performance! Loved every minute of it.", "age_group":
"18-25", "gender": "Male", "occupation": "Student"},
    {"rating": 3, "comments": "It was okay, could be better.", "age_group": "26-35", "gender":
"Female", "occupation": "Working Professional"},
    # ... other feedback responses
]
```

- Each **dictionary** contains:
  - **rating:** `int` (1 to 5, where 1 is the lowest rating and 5 is the highest rating)
  - **comments:** `str` (A string that contains the feedback comment)
  - **age\_group:** `str` (A string indicating the age group of the attendee, e.g., "18-25")
  - **gender:** `str` (A string indicating the gender of the attendee, e.g., "Male", "Female")
  - **occupation:** `str` (A string indicating the occupation of the attendee, e.g., "Student", "Working Professional")

---

## 2. demographics (Dictionary)

- **Type:** `dict`
- **Description:** A dictionary used to group feedback by demographic information, i.e., by `age_group`, `gender`, and `occupation`.
- **Example:**

```
demographics = {  
    ("18-25", "Male", "Student"): {"ratings": [5], "comments": ["Amazing performance! Loved  
every minute of it."]},  
    ("26-35", "Female", "Working Professional"): {"ratings": [3], "comments": ["It was okay,  
could be better."]},  
    # ... other demographic groups  
}
```

- **Key:** A tuple (`"age_group"`, `"gender"`, `"occupation"`) represents a unique demographic group.
- **Value:** A dictionary with two keys:
  - **ratings:** A list of `int` values representing the ratings given by attendees in that demographic group.
  - **comments:** A list of `str` values containing the feedback comments from attendees in that demographic group.

---

## 3. averages (Dictionary)

- **Type:** `dict`
- **Description:** A dictionary where the keys are demographic groups (tuples of `age_group`, `gender`, `occupation`), and the values are the average ratings for each group.
- **Example:**

```
averages = {  
    ("18-25", "Male", "Student"): 5.0,  
    ("26-35", "Female", "Working Professional"): 3.0,  
    # ... other demographic group average ratings  
}
```

- **Key:** A tuple (`"age_group"`, `"gender"`, `"occupation"`)
  - **Value:** A `float` representing the average rating for that demographic group.
-

#### 4. **positive\_keywords** and **negative\_keywords** (Dictionaries)

- **Type:** `dict`
- **Description:** These are dictionaries used to count the occurrences of specific positive or negative words in feedback comments.
- **Example:**

```
positive_keywords = {  
    "amazing": 1,  
    "fantastic": 1,  
    "great": 1,  
    "loved": 1,  
    "excellent": 1,  
    "awesome": 0,  
}  
negative_keywords = {  
    "terrible": 1,  
    "disappointing": 1,  
    "could be better": 1,  
    "worst": 0,  
    "awful": 0,  
}
```

- **Key:** A string representing the positive or negative word.
  - **Value:** An `int` representing the count of how many times that word appeared in the feedback comments.
- 

#### 5. **positive\_count** and **negative\_count** (Integers)

- **Type:** `int`
- **Description:** These are integer variables used to store the total number of responses categorized as positive or negative based on the ratings (e.g., ratings 4 and 5 are considered positive).
- **Example:**

```
positive_count = 4  
negative_count = 2
```

---

#### 6. **rating** (Integer)

- **Type:** `int`

- **Description:** Represents the rating given by an individual in the feedback response (from 1 to 5).
- **Example:**

rating = 5

---

## 7. **comment** (String)

- **Type:** `str`
- **Description:** Represents the textual comment given by the attendee in the feedback response.
- **Example:**

comment = "Amazing performance! Loved every minute of it."

---

## 8. **word** (String)

- **Type:** `str`
- **Description:** Represents individual words that are being searched in feedback comments for either positive or negative keywords.
- **Example:**

word = "amazing"

---

## 9. **count** (Integer)

- **Type:** `int`
- **Description:** Represents the count of how many times a specific word appears in the comments.
- **Example:**

count = 1

---

## Summary of Data Types:

- **List:** Used for storing multiple feedback responses and for tracking multiple ratings/comments.
- **Dictionary:** Used for categorizing feedback by demographic groups, counting occurrences of positive/negative words, and storing average ratings.



- **Tuple:** Used as the key for demographic groups in dictionaries.
  - **String:** Used for storing textual comments, age group, gender, and occupation.
  - **Integer:** Used for ratings and counting occurrences of specific words.
  - **Float:** Used for calculating average ratings.
- 

This approach allows for efficiently analyzing feedback in a multiplex society, where demographic factors influence how people perceive and rate events. The data types are designed to handle both structured and unstructured data (like ratings and comments) in a straightforward manner.