```
# Task-3
# College Event Feedback Analysis
```

```
from google.colab import files
uploaded=files. upload()
print(uploaded)
```

Choose Files student dataset.csv

- **student dataset.csv**(text/csv) - 25879 bytes, last modified: 10/2/2025 - 100% done

```
Saving student dataset.csv to student dataset.csv
{'student dataset.csv': b',Student ID,Well versed with the subject,Explains concepts in an understandable way,Use of presentati
```

```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
from textblob import TextBlob

# -----------------------------
# 1. Load Dataset
# -----------------------------
file_path = "student dataset.csv"
df = pd.read_csv(file_path)

print("Columns in dataset:", df.columns.tolist())
print(df.head())

# -----------------------------
# 2. Ratings Analysis
# -----------------------------
# Auto-detect numeric rating columns
rating_cols = df.select_dtypes(include=["int64","float64"]).columns.tolist()
print("\nDetected Rating Columns:", rating_cols)

# Overall averages
avg_ratings = df[rating_cols].mean()
print("\nAverage Ratings:")
print(avg_ratings)

# Plot bar chart of average ratings
avg_ratings.plot(kind="bar", figsize=(8,5), color="skyblue", title="Average Ratings (Overall)")
plt.ylabel("Average Score")
plt.xticks(rotation=45)
plt.show()

# If dataset contains "Event" column, also group by it
if "Event" in df.columns:
    event_avg = df.groupby("Event")[rating_cols].mean()
    event_avg.plot(kind="bar", figsize=(10,6), title="Average Ratings by Event")
    plt.ylabel("Average Score")
    plt.xticks(rotation=45)
    plt.show()

# -----------------------------
# 3. Sentiment Analysis
# -----------------------------
if "Feedback" in df.columns:
    df["Sentiment"] = df["Feedback"].astype(str).apply(lambda x: TextBlob(x).sentiment.polarity)

    print("\nSentiment Summary:")
    print(df["Sentiment"].describe())

    # Classify sentiment
    df["SentimentLabel"] = df["Sentiment"].apply(
        lambda x: "Positive" if x > 0.1 else ("Negative" if x < -0.1 else "Neutral")
    )
    print("\nSentiment Distribution:")
    print(df["SentimentLabel"].value_counts())

    # Pie chart
    df["SentimentLabel"].value_counts().plot(
        kind="pie", autopct="%1.1f%%", figsize=(6,6), title="Feedback Sentiment Distribution"
    )
    plt.ylabel("")
    plt.show()

    # Word cloud
    all_feedback = " ".join(df["Feedback"].dropna().astype(str))
    wordcloud = WordCloud(width=800, height=400, stopwo✦ STOPWORDS, background_color="white").generate(all_feedback)
```

```
        plt.figure(figsize=(10,5))
        plt.imshow(wordcloud, interpolation="bilinear")
        plt.axis("off")
        plt.title("Common Feedback Themes")
        plt.show()

    # ----------------------------
    # 4. Key Recommendations
    # ----------------------------
    print("\n🔑 Key Recommendations for Organizers:")

    # Find weakest rated areas
    weak_areas = avg_ratings.sort_values().head(2).index.tolist()
    for area in weak_areas:
        if "Logistics" in area:
            print("- Improve event logistics: registration, seating, sound system.")
        elif "Content" in area:
            print("- Enhance event content: align topics with student interests.")
        elif "Speaker" in area:
            print("- Invite engaging speakers and encourage interaction.")
        elif "Satisfaction" in area:
            print("- Gather real-time feedback during events to boost satisfaction.")
        else:
            print(f"- Focus on improving {area}.")
```

```
Columns in dataset: ['Unnamed: 0', 'Student ID', 'Well versed with the subject', 'Explains concepts in an understandable way',
   Unnamed: 0  Student ID  Well versed with the subject  \
0           0         340                             5
1           1         253                             6
2           2         680                             7
3           3         806                             9
4           4         632                             8

   Explains concepts in an understandable way  Use of presentations  \
0                                           2                     7
1                                           5                     8
2                                           7                     6
3                                           6                     7
4                                          10                     8

   Degree of difficulty of assignments  Solves doubts willingly  \
0                                    6                        9
1                                    6                        2
2                                    5                        4
3                                    1                        5
4                                    4                        6

   Structuring of the course  \
0                          2
1                          1
2                          2
3                          9
4                          6

   Provides support for students going above and beyond  \
0                                                  1
1                                                  2
2                                                  3
3                                                  4
4                                                  9

   Course recommendation based on relevance
0                                         8
1                                         9
2                                         1
3                                         6
4                                         9
```

```
Start coding or generate with AI.
```

```
Average Ratings:
Unnamed: 0                                             500.000000
Student ID                                             500.000000
Well versed with the subject                             7.497502
Explains concepts in an understandable way               6.081918
Use of presentations                                     5.942058
Degree of difficulty of assignments                      5.430569
Solves doubts willingly                                  5.474525
Structuring of the course                                5.636364
Provides support for students going above and beyond     5.662338
Course recommendation based on relevance                 5.598402
dtype: float64
```