📊 College Event Feedback Analysis (CSV Version)

This notebook analyzes the newly uploaded CSV feedback data with visualizations and insights. *Includes bar chart, heatmap, pie chart, and word cloud.*

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from wordcloud import WordCloud
from google.colab import files
from textblob import TextBlob

sns.set(style="whitegrid")
```

```
uploaded = files.upload()
df = pd.read_csv("Cleaned_Student_Feedback.csv")
df.head()
```

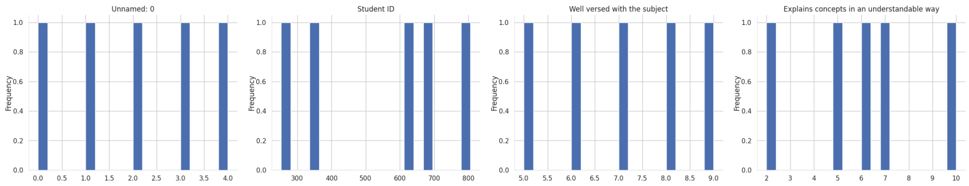
Choose Files Cleaned_St...edback.csv

- **Cleaned_Student_Feedback.csv**(text/csv) - 24877 bytes, last modified: 8/3/2025 - 100% done

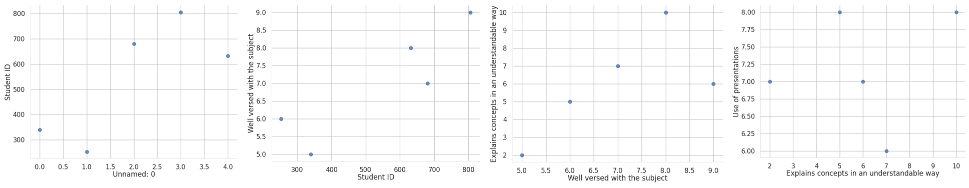
```
Saving Cleaned_Student_Feedback.csv to Cleaned_Student_Feedback (3).csv
```

	Unnamed: 0	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuring of the course	Provides support for students going above and beyond	Course recommendation based on relevance
0	0	340	5	2	7	6	9	2	1	8
1	1	253	6	5	8	6	2	1	2	9
2	2	680	7	7	6	5	4	2	3	1
3	3	806	9	6	7	1	5	9	4	6
4	4	632	8	10	8	4	6	6	9	9

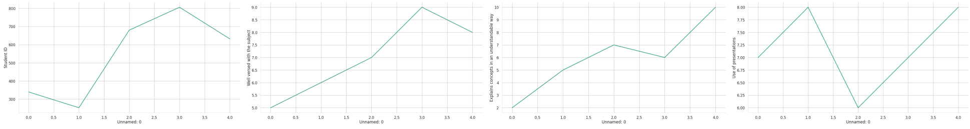
Distributions



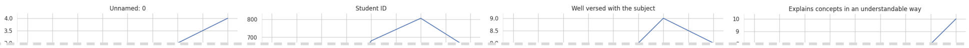
2-d distributions



Time series



Values



Next steps: Generate code with df View recommended plots New interactive sheet

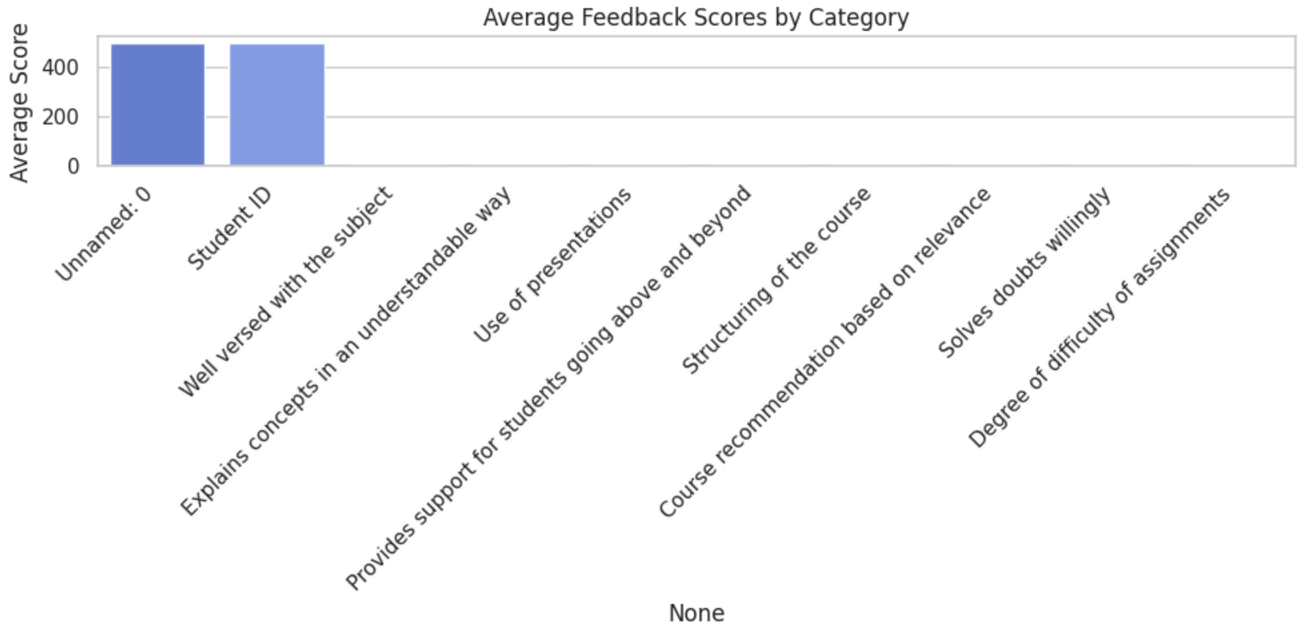
```
average_scores = df.mean().sort_values(ascending=False)

plt.figure(figsize=(10, 5))
sns.barplot(x=average_scores.index, y=average_scores.values, palette='coolwarm')
plt.title("Average Feedback Scores by Category")
plt.ylabel("Average Score")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

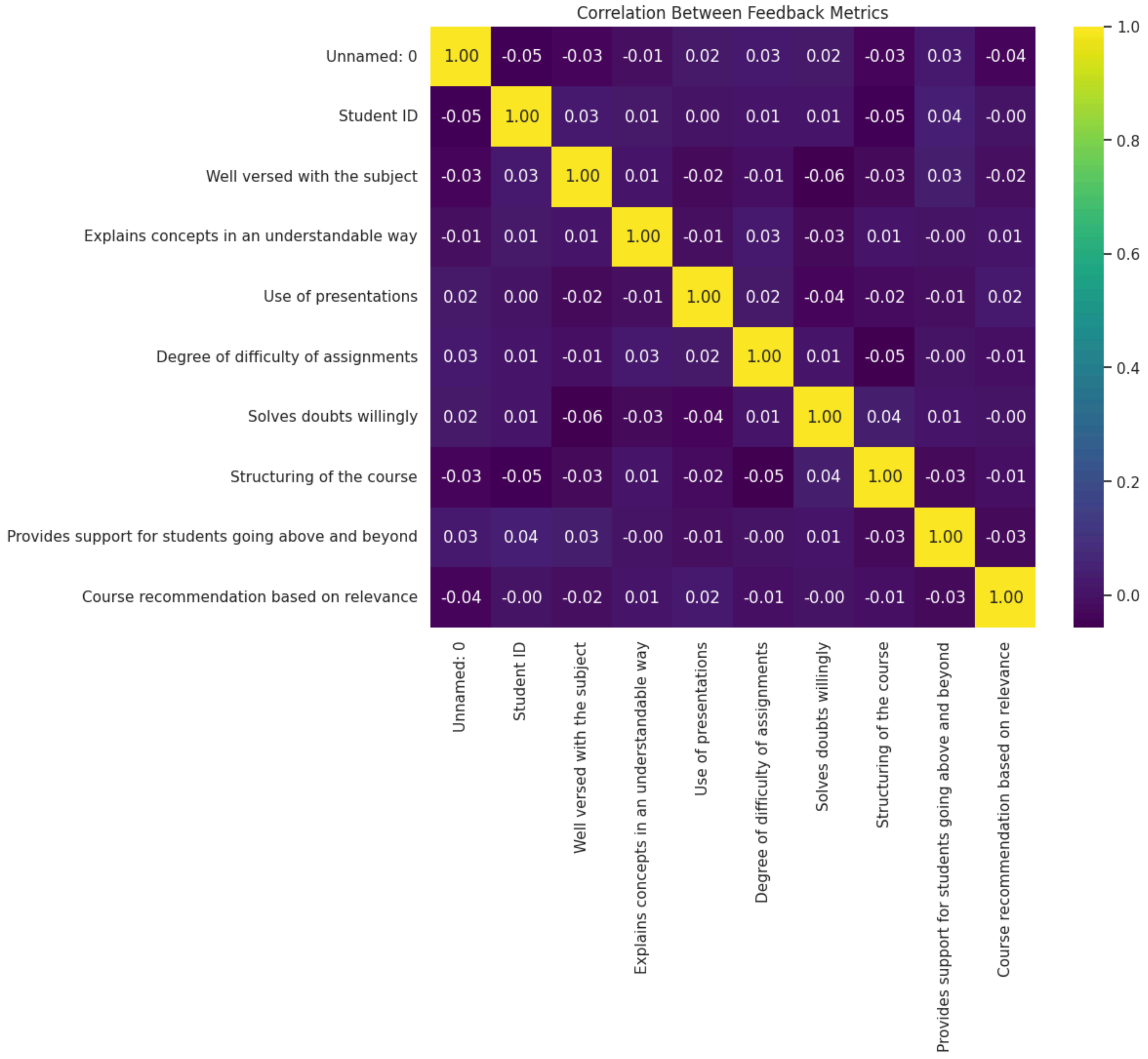
```
/tmp/ipython-input-955647672.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.barplot(x=average_scores.index, y=average_scores.values, palette='coolwarm')
```



```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='viridis', fmt=".2f")
plt.title("Correlation Between Feedback Metrics")
plt.show()
```

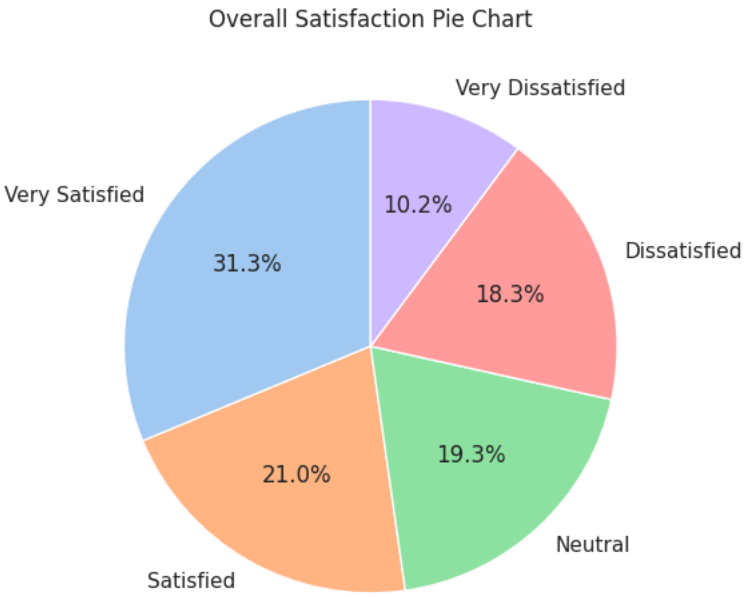

Correlation Between Feedback Metrics

	Unnamed: 0	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuring of the course	Provides support for students going above and beyond	Course recommendation based on relevance
Unnamed: 0	1.00	-0.05	-0.03	-0.01	0.02	0.03	0.02	-0.03	0.03	-0.04
Student ID	-0.05	1.00	0.03	0.01	0.00	0.01	0.01	-0.05	0.04	-0.00
Well versed with the subject	-0.03	0.03	1.00	0.01	-0.02	-0.01	-0.06	-0.03	0.03	-0.02
Explains concepts in an understandable way	-0.01	0.01	0.01	1.00	-0.01	0.03	-0.03	0.01	-0.00	0.01
Use of presentations	0.02	0.00	-0.02	-0.01	1.00	0.02	-0.04	-0.02	-0.01	0.02
Degree of difficulty of assignments	0.03	0.01	-0.01	0.03	0.02	1.00	0.01	-0.05	-0.00	-0.01
Solves doubts willingly	0.02	0.01	-0.06	-0.03	-0.04	0.01	1.00	0.04	0.01	-0.00
Structuring of the course	-0.03	-0.05	-0.03	0.01	-0.02	-0.05	0.04	1.00	-0.03	-0.01
Provides support for students going above and beyond	0.03	0.04	0.03	-0.00	-0.01	-0.00	0.01	-0.03	1.00	-0.03
Course recommendation based on relevance	-0.04	-0.00	-0.02	0.01	0.02	-0.01	-0.00	-0.01	-0.03	1.00

```
def label_satisfaction(val):
    if val >= 8:
        return "Very Satisfied"
    elif val >= 6:
        return "Satisfied"
    elif val >= 4:
        return "Neutral"
    elif val >= 2:
        return "Dissatisfied"
    else:
        return "Very Dissatisfied"

df["Satisfaction Level"] = df["Course recommendation based on relevance"].apply(label_satisfaction)
satisfaction_counts = df["Satisfaction Level"].value_counts()

plt.figure(figsize=(6, 6))
satisfaction_counts.plot.pie(autopct='%1.1f%%', colors=sns.color_palette("pastel"), startangle=90)
plt.title("Overall Satisfaction Pie Chart")
plt.ylabel("")
plt.show()
```

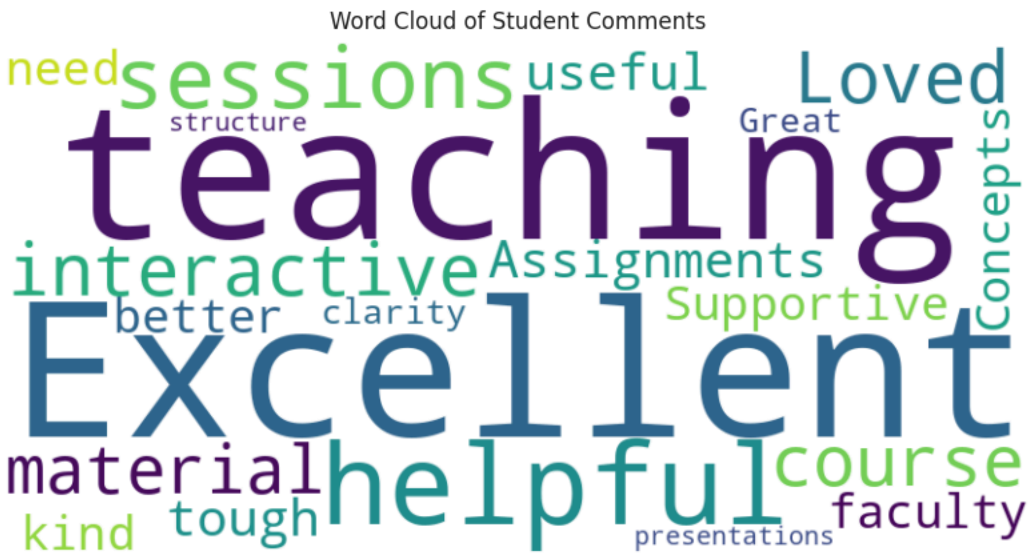


```
sample_feedback = [
    "Excellent teaching and helpful sessions.",
    "Could be more interactive.",
    "Loved the course material.",
    "Assignments were tough but useful.",
    "Supportive and kind faculty.",
    "Concepts need better clarity.",
    "Great presentations and structure."
]

text_blob = " ".join(sample_feedback)

wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text_blob)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Word Cloud of Student Comments")
plt.show()
```


Word Cloud of Student Comments
need
sessions
useful
Loved
structure
Great
teaching
Concepts
interactive
Assignments
better
clarity
Supportive
Excellent
material
helpful
course
kind
tough
presentations
faculty