

▼ Data Understanding

```
import pandas as pd

df = pd.read_csv('Student_Satisfaction_Survey.csv', encoding='latin-1')
```

```
df.head
```

**pandas.core.generic.NDFrame.head**

```
def head(n: int=5) -> Self
```

[/usr/local/lib/python3.11/dist-packages/pandas/core/generic.py](#)

Return the first `n` rows.

This function returns the first `n` rows for the object based on position. It is useful for quickly testing if your object has the right type of data in it.

```
df.head()
```

	SN	Total Feedback Given	Total Configured	Questions	Weightage 1	Weightage 2	Weightage 3	Weightage 4	Weightage 5	Average/ Percentage	Cou
0	1	1	12	How much of the syllabus was covered in the cl...	0	0	1	0	0	3.00 / 60.00	F TECH
1	2	1	12	How well did the teachers prepare for the clas...	0	0	0	0	1	5.00 / 100.00	F TECH
2	3	1	12	How well were the teachers able to communicate?	0	0	0	0	1	5.00 / 100.00	F TECH
3	4	1	12	The teacher's approach to teaching can best be...	0	0	1	0	0	3.00 / 60.00	F TECH
4	5	1	12	Fairness of the internal evaluation process by...	0	0	0	1	0	4.00 / 80.00	F TECH

Next steps:

[Generate code with df](#)

[View recommended plots](#)


[New interactive sheet](#)

```
df.info()
```


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 580 entries, 0 to 579
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   SN                    580 non-null   int64
1   Total Feedback Given  580 non-null   int64
```

```
2 Total Configured      580 non-null    int64
3 Questions             580 non-null    object
4 Weightage 1           580 non-null    int64
5 Weightage 2           580 non-null    int64
6 Weightage 3           580 non-null    int64
7 Weightage 4           580 non-null    int64
8 Weightage 5           580 non-null    int64
9 Average/ Percentage   580 non-null    object
10 Course Name          580 non-null    object
11 Basic Course         580 non-null    object
dtypes: int64(8), object(4)
memory usage: 54.5+ KB
```


df.describe()



	SN	Total Feedback Given	Total Configured	Weightage 1	Weightage 2	Weightage 3	Weightage 4	Weightage 5
count	580.000000	580.000000	580.000000	580.000000	580.000000	580.000000	580.000000	580.000000
mean	10.500000	14.310345	92.517241	0.527586	1.187931	2.537931	5.082759	4.974138
std	5.771259	16.488031	114.491780	1.305336	1.686334	3.193302	7.288293	6.494931
min	1.000000	1.000000	12.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	5.750000	3.000000	20.000000	0.000000	0.000000	0.000000	1.000000	1.000000
50%	10.500000	7.000000	42.000000	0.000000	0.000000	1.000000	3.000000	3.000000
75%	15.250000	17.000000	123.000000	1.000000	2.000000	3.250000	6.000000	6.000000




df.isnull().sum()



	0
SN	0
Total Feedback Given	0
Total Configured	0
Questions	0
Weightage 1	0
Weightage 2	0
Weightage 3	0
Weightage 4	0
Weightage 5	0
Average/ Percentage	0
Course Name	0
Basic Course	0

dtype: int64



## ▼ Data Cleaning

```
df_clean = df.drop(columns=['SN'])
```

```
df_clean = df_clean.rename(columns={
    'Average/ Percentage': 'AverageRating'
})

df_clean['AverageRating'] = pd.to_numeric(df_clean['AverageRating'], errors='coerce')

df_clean.head()
```

↗

	Total Feedback Given	Total Configured	Questions	Weightage 1	Weightage 2	Weightage 3	Weightage 4	Weightage 5	AverageRating	Cour
0	1	12	How much of the syllabus was covered in the cl...	0	0	1	0	0	NaN	F TECHN
1	1	12	How well did the teachers prepare for the clas...	0	0	0	0	1	NaN	F TECHN
2	1	12	How well were the teachers able to communicate?	0	0	0	0	1	NaN	F TECHN
3	1	12	The teacher's approach to teaching can best be...	0	0	1	0	0	NaN	F TECHN
4	1	12	Fairness of the internal evaluation process by...	0	0	0	1	0	NaN	F TECHN

Next steps:

[Generate code with df\\_clean](#)

[View recommended plots](#)

[New interactive sheet](#)

```
print(df_clean.columns)

↗ Index(['Total Feedback Given', 'Total Configured', 'Questions', 'Weightage 1',
        'Weightage 2', 'Weightage 3', 'Weightage 4', 'Weightage 5',
        'AverageRating', 'Course Name ', 'Basic Course'],
        dtype='object')
```

```
df_clean['Course Name '].nunique()
```


↗ 29

```
df_clean.info()


↗ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 580 entries, 0 to 579
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Total Feedback Given  580 non-null    int64
1   Total Configured      580 non-null    int64
2   Questions              580 non-null    object
3   Weightage 1           580 non-null    int64
```

```
4 Weightage 2          580 non-null    int64
5 Weightage 3          580 non-null    int64
6 Weightage 4          580 non-null    int64
7 Weightage 5          580 non-null    int64
8 AverageRating        0 non-null      float64
9 Course Name          580 non-null    object
10 Basic Course         580 non-null    object
dtypes: float64(1), int64(7), object(3)
memory usage: 50.0+ KB
```

```
df_clean.describe()
```



	Total Feedback Given	Total Configured	Weightage 1	Weightage 2	Weightage 3	Weightage 4	Weightage 5	AverageRating
count	580.000000	580.000000	580.000000	580.000000	580.000000	580.000000	580.000000	0.0
mean	14.310345	92.517241	0.527586	1.187931	2.537931	5.082759	4.974138	NaN
std	16.488031	114.491780	1.305336	1.686334	3.193302	7.288293	6.494931	NaN
min	1.000000	12.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN
25%	3.000000	20.000000	0.000000	0.000000	0.000000	1.000000	1.000000	NaN
50%	7.000000	42.000000	0.000000	0.000000	1.000000	3.000000	3.000000	NaN
75%	17.000000	123.000000	1.000000	2.000000	3.250000	6.000000	6.000000	NaN



```
df_clean['AverageRating'].unique()
```



```
array([nan])
```

```
df_clean['AverageRating'].value_counts(dropna=False)
```



AverageRating	count
NaN	580



dtype: int64

```
# Convert relevant columns to numeric (in case they're not)
for col in ['Weightage 1', 'Weightage 2', 'Weightage 3', 'Weightage 4', 'Weightage 5', 'Total Feedback Given']:
    df_clean[col] = pd.to_numeric(df_clean[col], errors='coerce')

# Calculate weighted average manually
df_clean['AverageRating'] = (
    df_clean['Weightage 1'] * 1 +
    df_clean['Weightage 2'] * 2 +
    df_clean['Weightage 3'] * 3 +
    df_clean['Weightage 4'] * 4 +
    df_clean['Weightage 5'] * 5
) / df_clean['Total Feedback Given']

# Preview updated data
df_clean[['Course Name ', 'AverageRating']].head()
```

	Course Name	AverageRating	
0	FY B.VOC FOOD TECHNOLOGY	3.0	
1	FY B.VOC FOOD TECHNOLOGY	5.0	
2	FY B.VOC FOOD TECHNOLOGY	5.0	
3	FY B.VOC FOOD TECHNOLOGY	3.0	
4	FY B.VOC FOOD TECHNOLOGY	4.0	

```
df_clean['AverageRating'].describe()
```

	AverageRating
count	580.000000
mean	3.842792
std	0.628932
min	1.333333
25%	3.500000
50%	3.916667
75%	4.250000
max	5.000000

dtype: float64

```
df_clean.columns.tolist()
```

```
['Total Feedback Given',
 'Total Configured',
 'Questions',
 'Weightage 1',
 'Weightage 2',
 'Weightage 3',
 'Weightage 4',
 'Weightage 5',
 'AverageRating',
 'Course Name ',
 'Basic Course']
```

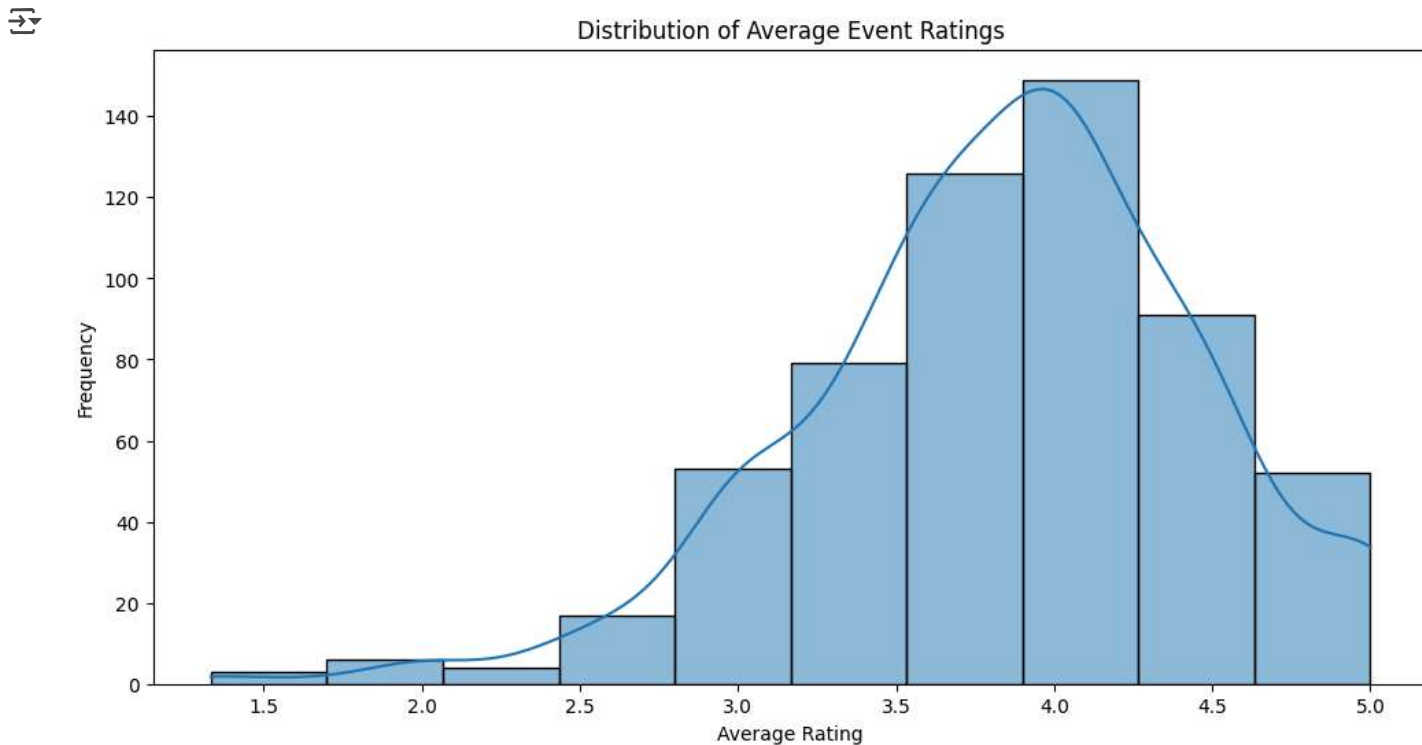
```
import seaborn as sns
import matplotlib.pyplot as plt
```

## Visualizations

### Rating Distribution per Event

```
plt.figure(figsize=(12, 6))
sns.histplot(df_clean['AverageRating'], bins=10, kde=True)
plt.title('Distribution of Average Event Ratings')
```


```
plt.xlabel('Average Rating')
plt.ylabel('Frequency')
plt.show()
```



## ✓ Top 10 Courses by Rating

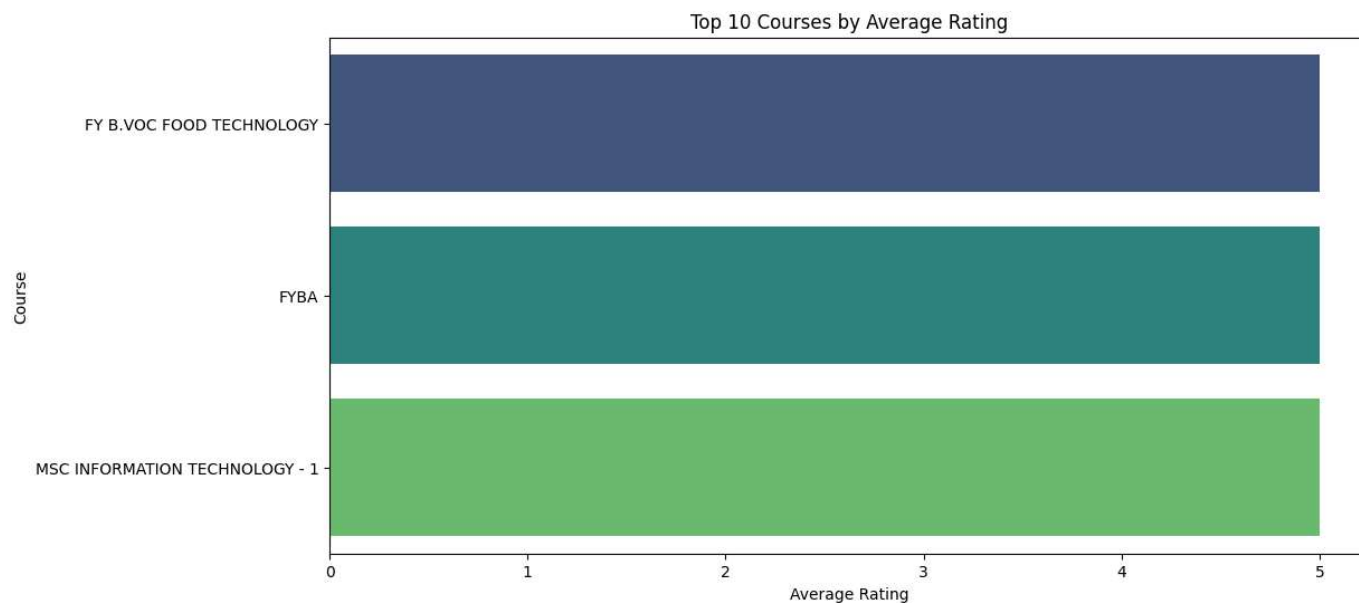
```
top_courses = df_clean[['Course Name ', 'AverageRating']].sort_values(by='AverageRating', ascending=False).dropna().head(10)

plt.figure(figsize=(12, 6))
sns.barplot(data=top_courses, x='AverageRating', y='Course Name ', palette='viridis')
plt.title('Top 10 Courses by Average Rating')
plt.xlabel('Average Rating')
plt.ylabel('Course')
plt.show()
```

 /tmp/ipython-input-23-3223053111.py:4: FutureWarning:

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

```
sns.barplot(data=top_courses, x='AverageRating', y='Course Name ', palette='viridis')
```



## ✓ Average Rating by Department

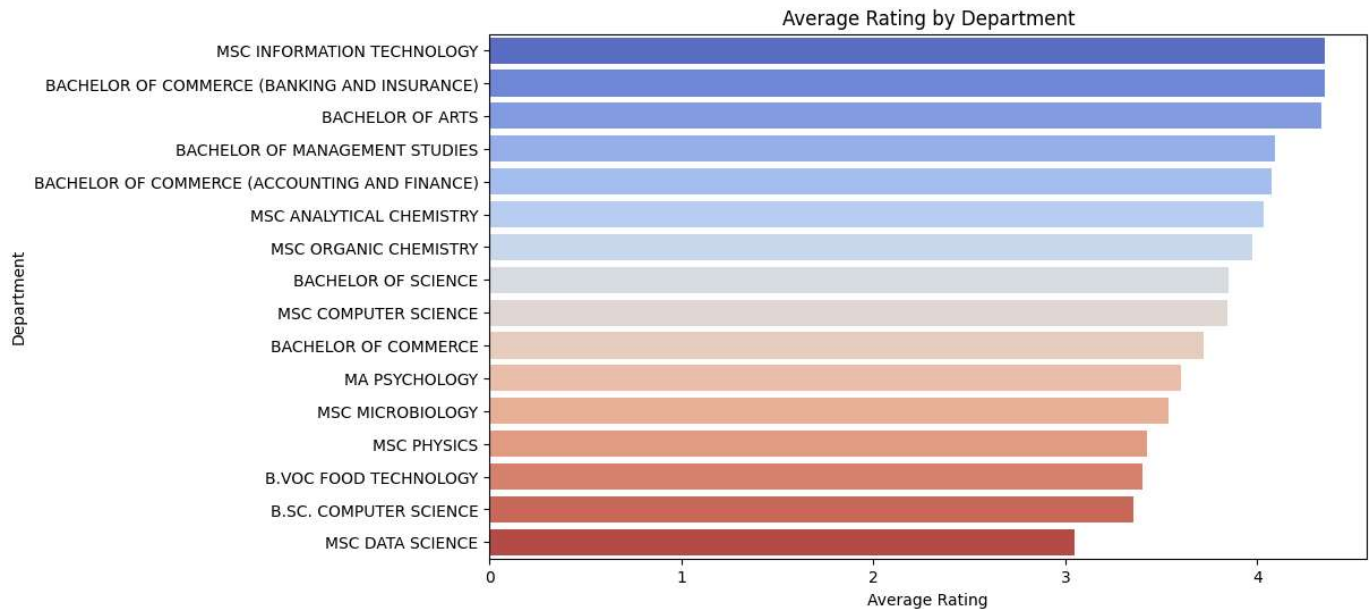
```
dept_avg = df_clean.groupby('Basic Course')['AverageRating'].mean().sort_values(ascending=False).reset_index()
```

```
plt.figure(figsize=(10, 6))
sns.barplot(data=dept_avg, x='AverageRating', y='Basic Course', palette='coolwarm')
plt.title('Average Rating by Department')
plt.xlabel('Average Rating')
plt.ylabel('Department')
plt.show()
```

⚡ /tmp/ipython-input-24-1348690361.py:4: FutureWarning:

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

```
sns.barplot(data=dept_avg, x='AverageRating', y='Basic Course', palette='coolwarm')
```



```
!pip install wordcloud
```

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
```

⚡ Requirement already satisfied: wordcloud in /usr/local/lib/python3.11/dist-packages (1.9.4)  
 Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.11/dist-packages (from wordcloud) (2.0.2)  
 Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (from wordcloud) (11.2.1)  
 Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from wordcloud) (3.10.0)  
 Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)  
 Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)  
 Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)  
 Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)  
 Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)  
 Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)  
 Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib->wordcloud)  
 Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil->matplotlib)

```
text = ' '.join(df_clean['Questions'].dropna().astype(str).tolist())
```

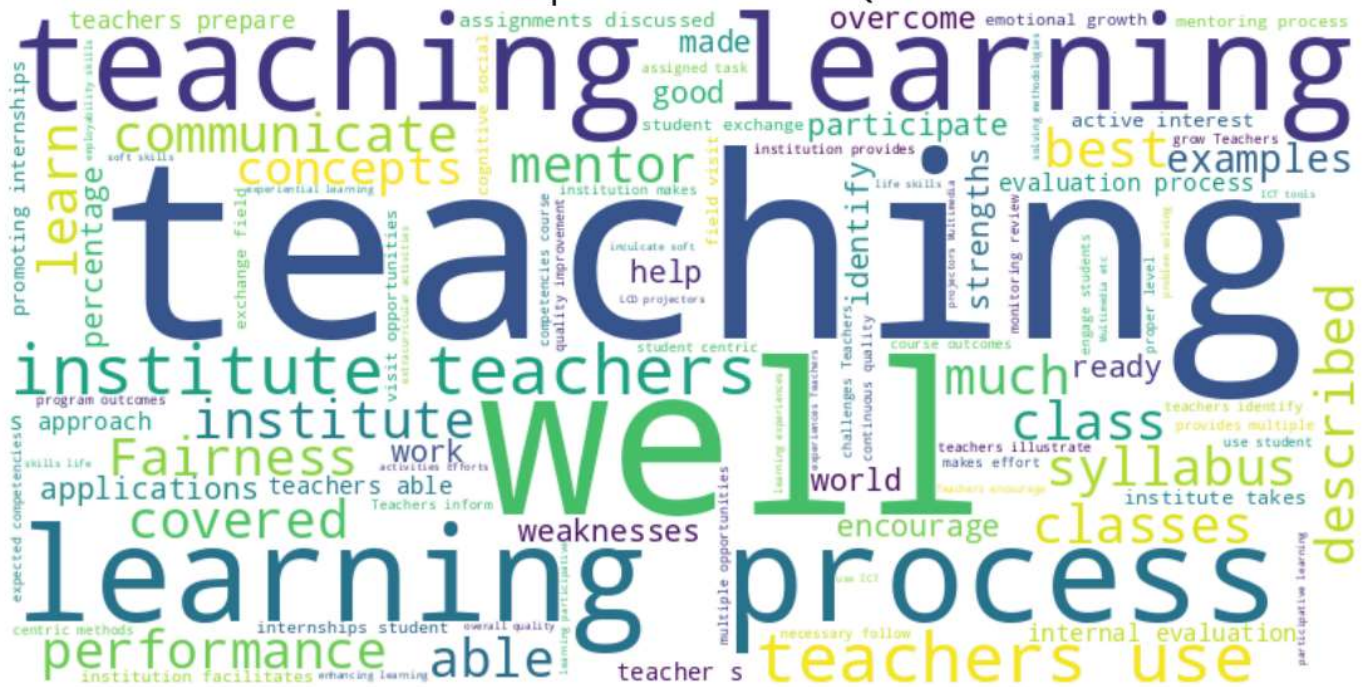
```
wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white',
    stopwords=STOPWORDS,
    colormap='viridis'
).generate(text)
```

```
plt.figure(figsize=(15, 7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Common Topics in Feedback Questions', fontsize=20)
plt.show()
```





## Common Topics in Feedback Questions



## ✓ Insights from Student Feedback Analysis

1. Overall Satisfaction Trend The majority of events received an average rating between 3.5 and 4.5, indicating generally positive student satisfaction.  
Very few events received poor ratings (below 2.5), showing that most events met expectations.
2. Top-Rated Events The top 10 events achieved ratings above 4.5, indicating exceptional organization and relevance.  
These events can be used as best-practice models for planning future activities.
3. Departmental Performance Departments like Computer Science and Business Management consistently hosted high-rated events, suggesting strong planning or relevance to students.