MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Department Of Computer Engineering

B.TECH SEM- I I

NAME :	Abhijeet Dnyaneshwar Damal
SUBJECT:	Essentials Of Data-Science (EDS)
DIVISION:	CS4
BATCH :	C42
ROLL NO.:	40
PRN No. :	202401040307

Under the Guidance of Course In-charge,

Prof. Poonam Manjare

Theory

Activity No. 01-

Objective:

To Formulate 20 problem statements for a given dataset using Numpy and Pandas and Apply Numpy and pandas methods to find the solution for the formulated problem statements.

Dataset Name:

Yelp Reviews

Dataset link :

https://www.kaggle.com/datasets/omkarsabnis/yelp-reviews-dataset

Google Colab Link for the activity:

https://colab.research.google.com/drive/16tln35L0uYX098XFZGPySlnXgpgVYZkg?usp=sharing

.ipynb File to open in VS code or any other Editor:

Yelp_Reviews_Activity_No_1.ipynb

Final_Numpy_Pandas_Assignment.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text

RAM Disk

```
[2] import pandas as pd
    import numpy as np
```

```
df = pd.read_csv('/content/yelp.csv')
df
```

	business_id	date	review_id	stars	text	type	user_id	cool	useful	funny
0	9yKzy9PApeiPPOUJEtnvkg	2011-01-26	fWKvX83p0-ka4JS3dc6E5A	5	My wife took me here on my birthday for breakf...	review	rLtl8ZkDX5vH5nAx9C3q5Q	2	5	0
1	ZRJwVLyzEJq1VAihDhYiow	2011-07-27	IjZ33sJrzXqU-0X6U8NwyA	5	I have no idea why some people give bad review...	review	0a2KyEL0d3Yb1V6aivbIuQ	0	0	0
2	6oRAC4uyJCsJl1X0WZpVSA	2012-06-14	IESLBzqUCLdSzSqm0eCSxQ	4	love the gyro plate. Rice is so good and I als...	review	0hT2KtfLiobPvh6cDC8JQg	0	1	0
3	_1QQZuf4zZOyFCvXc0o6Vg	2010-05-27	G-WvGaISbqqaMHlNnByodA	5	Rosie, Dakota, and I LOVE Chaparral Dog Park!!...	review	uZetl9T0NcROGOyFfughhg	1	2	0
4	6ozycU1RpktNG2-1BroVtw	2012-01-05	1uJFq2r5QfJG_6ExMRCaGw	5	General Manager Scott Petello is a good egg!!!...	review	vYmM4KTsC8ZfQBg-j5MWkw	0	0	0
...	...	...	...	...	...	...	...	...	...	...
9995	VY_tvNUCCXGXQeSvJl757Q	2012-07-28	Ubyfp2RSDYW0g7Mbr8N3iA	3	First visit...Had lunch here today - used my G...	review	_eqQoPtQ3e3UxLE4faT6ow	1	2	0
9996	EKzMHI1tip8rC1-ZAy64yg	2012-01-18	2XyIOQKbVFb6uXQdJ0RzlQ	4	Should be called house of deliciousness!\n\nI ...	review	ROru4uk5SaYc3rg8IU7SQw	0	0	0
9997	53YGfwmbW73JhFiemNeyzQ	2010-11-16	jyznYkIbpqVmlsZxSDSypA	4	I recently visited Olive and Ivy for business ...	review	gGbN1aKQHMgfQZkqlsuwzg	0	0	0
9998	9SKdOoDHcFoxK5ZtsgHJoA	2012-12-02	5UKq9WQE1qQbJ0DJbc-B6Q	2	My nephew just moved to Scottsdale recently so...	review	0lyVoNazXa20WzUyZPLaQQ	0	0	0
9999	pF7uRzygyZsltbmVpjIyvw	2010-10-16	vWSmOhg2ID1MNZHaWapGbA	5	4-5 locations.. all 4.5 star average.. I think...	review	KSBFytcdjPKZgXKQnYQdkA	0	0	0

10000 rows × 10 columns

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

1. Find the average rating from the 'stars' column.

```
[39] stars = df['stars'].to_numpy()
     avg_rating = np.mean(stars)
     print('Average Rating:', avg_rating)
```

```
Average Rating: 3.7775
```

2. Find the maximum number of 'useful' votes received by any review.

```
[41] useful_votes = df['useful'].to_numpy()
     max_useful = np.max(useful_votes)
     print('Max Useful Votes:', max_useful)
```

```
Max Useful Votes: 76
```

3.Calculate the standard deviation of 'funny' votes.

```
[42] funny_votes = df['funny'].to_numpy()
     std_funny = np.std(funny_votes)
     print('Standard Deviation of Funny Votes:', std_funny)
```

```
Standard Deviation of Funny Votes: 1.9078465111218983
```

4. Find how many reviews received exactly 5 'cool' votes.

```
[43] cool_votes = df['cool'].to_numpy()
     count_cool_5 = np.sum(cool_votes == 5)
     print('Number of Reviews with Exactly 5 Cool Votes:', count_cool_5)
```

```
Number of Reviews with Exactly 5 Cool Votes: 119
```

5. Calculate the percentage of reviews that have 0 'useful' votes.

5. Calculate the percentage of reviews that have 0 'useful' votes.

```
[44] percentage_zero_useful = np.sum(useful_votes == 0) / useful_votes.size * 100
     print('Percentage of Reviews with 0 Useful Votes:', np.round(percentage_zero_useful, 2),'%')
```

```
Percentage of Reviews with 0 Useful Votes: 41.3 %
```

6.Find the median value of 'stars'.

```
[45] median_stars = np.median(stars)
     print('Median Stars:', median_stars)
```

```
Median Stars: 4.0
```

7.Find how many reviews have 'funny' votes greater than average 'funny' votes.

```
[46] avg_funny = np.mean(funny_votes)
     count_above_avg_funny = np.sum(funny_votes > avg_funny)
     print('Reviews with Funny Votes Above Average:', count_above_avg_funny)
```

```
Reviews with Funny Votes Above Average: 2987
```

8. The percentage of reviews that received a 1-star rating.

```
print("The percentage of reviews that received a 1-star rating")
print((df['stars'] == 1).mean() * 100)
```

```
The percentage of reviews that received a 1-star rating
7.489999999999999
```

9. Find the total sum of 'funny' votes.

```
[49] total_funny_votes = np.sum(funny_votes)
     print('Total Funny Votes:', total_funny_votes)
```

```
Total Funny Votes: 7013
```

10. Find how many reviews have 'cool' votes equal to 'funny' votes.

```
[51] count_equal_cool_funny = np.sum(cool_votes == funny_votes)
     print('Reviews where Cool Votes Equal Funny Votes:', count_equal_cool_funny)
```

```
Reviews where Cool Votes Equal Funny Votes: 6790
```

11. Find the total number of reviews.

```
[20] total_reviews = df.shape[0]
     print('Total Number of Reviews:', total_reviews)
```

```
Total Number of Reviews: 10000
```

12. Find the number of unique users who have written reviews.

```
[52] unique_users = df['user_id'].nunique()
     print('Unique Users:', unique_users)
```

```
Unique Users: 6403
```

13. Calculate the average number of words per review.

```
[53] avg_words = df['text'].str.split().apply(len).mean()
     print('Average Words Per Review:', avg_words)
```

```
Average Words Per Review: 131.0396
```

14. Identify the user who has written the most reviews.

```
[55] top_reviewer = df['user_id'].value_counts().idxmax()
     print('User with Most Reviews:', top_reviewer)
```

```
User with Most Reviews: fczQCSmaWF78toLEmb0Zsw
```

15. Count the number of reviews per year.

```
[60] print('Reviews per Year:')
     pd.to_datetime(df['date']).dt.year.value_counts().sort_index()
```

```
Reviews per Year:
```

date	count
2005	4
2006	55
2007	285
2008	765
2009	1171
2010	1852
2011	2791
2012	3025
2013	52

dtype: int64

16. The number of reviews for each star rating (star distribution).

```
[67] print("The number of reviews for each star rating:")
     df['stars'].value_counts().sort_index()
```

```
The number of reviews for each star rating:
```

stars	count
1	749
2	927
3	1461
4	3526
5	3337

dtype: int64

17. Find the business that has received the most 5-star reviews.

```
[56] top_5star_business = df[df['stars'] == 5]['business_id'].value_counts().idxmax()
     print('Business with Most 5-Star Reviews:', top_5star_business)
```

```
Business with Most 5-Star Reviews: JokKtdXU7zXHcr20Lrk29A
```

18. The total number of reviews that mention the word “delicious”.

```
[58] delicious=df['text'].str.contains("delicious", case=False).sum()
     print("Total Delicious Reviews:",delicious)
```

```
Total Delicious Reviews: 1176
```

19. The number of reviews with more than 10 total votes (sum of 'cool', 'useful', and 'funny').

```
[59] num_reveiws=(df[['cool', 'useful', 'funny']].sum(axis=1) > 10).sum()
     print("number of reviews with more than 10 total votes",num_reveiws)
```

```
number of reviews with more than 10 total votes 613
```

20. The earliest date on which a review was submitted.

```
[30] df['date'].min()
```

```
'2005-04-18'
```