

```
1 from google.colab import files
2 uploaded = files.upload()
3
```




Choose Files

deepseek\_vs\_chatgpt.csv

- **deepseek\_vs\_chatgpt.csv**(text/csv) - 2416971 bytes, last modified: 4/6/2025 - 100% done

Saving deepseek\_vs\_chatgpt.csv to deepseek\_vs\_chatgpt.csv

```
1 import pandas as pd
2
3 df = pd.read_csv('deepseek_vs_chatgpt.csv') # Use the correct file name
4 df.head() # Optional: preview the data
5
```




	Date	Month_Num	Weekday	AI_Platform	AI_Model_Version	Active_Users	New_Users	Churned_Users	Daily_Churn_Rate	Retention_Rate	.
0	2024-09-21	9	Saturday	ChatGPT	GPT-4-turbo	500000	25000	25000	0.05	0.95	
1	2024-09-21	9	Saturday	ChatGPT	GPT-4-turbo	500000	25000	25000	0.05	0.95	
2	2024-09-21	9	Saturday	ChatGPT	GPT-4-turbo	500000	25000	25000	0.05	0.95	
3	2024-09-21	9	Saturday	ChatGPT	GPT-4-turbo	500000	25000	25000	0.05	0.95	
4	2024-05-16	5	Thursday	DeepSeek	DeepSeek-Chat 1.5	1700000	170000	34000	0.02	0.95	

5 rows × 28 columns


```
1 #Importing NumPy and Creating Arrays
2 import numpy as np
3 users = df['Active_Users'].values
4 ratings = df['User_Rating'].values
5 durations = df['Session_Duration_sec'].values
```

```
1 # Fixed Type Arrays & Data Types
2 users.dtype # dtype('int64')
3 ratings.dtype # dtype('int64')
```




```
dtype('int64')
```

```
1 #Array Indexing and Slicing
2 print(users[0]) # First user count
3 print(users[0:5]) # First 5 user counts
4 print(users[-1]) # Last entry
```



```
500000
[ 500000  500000  500000  500000 1700000]
1700000
```

```
1 #Reshaping Arrays
2 reshaped = users[:10].reshape(2, 5) # 2 rows, 5 cols
3 print("Reshaped array (2x5):\n", reshaped)
```



```
Reshaped array (2x5):
[[ 500000  500000  500000  500000 1700000]
 [1700000 1700000 1700000 1700000 1700000]]
```

```
1 #Concatenation and Splitting
2 combined = np.concatenate([users[:5], users[-5:]])
3 print("Concatenated array:", combined)
4
5 split1, split2 = np.split(users[:10], 2)
6 print("First split:", split1)
7 print("Second split:", split2)
```

```

Concatenated array: [ 500000  500000  500000  500000 1700000 1700000 1700000 1700000 1700000
1700000]
First split: [ 500000  500000  500000  500000 1700000]
Second split: [1700000 1700000 1700000 1700000 1700000]

```

```

1 #Universal Functions
2 print("Square root of first 5 ratings:", np.sqrt(ratings[:5]))
3 print("Log of first 5 user counts:", np.log1p(users[:5]))
4 print("Exponential of first 5 durations:", np.exp(durations[:5]))

```

```

Square root of first 5 ratings: [2.          2.          1.73205081 2.23606798 2.23606798]
Log of first 5 user counts: [13.12236538 13.12236538 13.12236538 13.12236538 14.3461394 ]
Exponential of first 5 durations: [2.35385267e+17 2.64891221e+10 5.83461743e+14 6.56599691e+07
2.20264658e+04]

```

```

1 #Aggregations
2 print("Average rating:", np.mean(ratings))
3 print("Total active users:", np.sum(users))
4 print("Maximum session duration:", np.max(durations))
5 print("Median session duration:", np.median(durations))

```

```

Average rating: 4.3947
Total active users: 11962550000
Maximum session duration: 60
Median session duration: 27.0

```

```

1 #Broadcasting Rules
2 normalized = durations / np.max(durations)
3 print("Normalized durations:", normalized[:5])

```

```

Normalized durations: [0.66666667 0.4          0.56666667 0.3          0.16666667]

```

```

1 #Comparisons and Boolean Arrays
2 high_rating = ratings > 4
3 long_sessions = durations > 60
4 print("High ratings (>4):", high_rating[:5])
5 print("Long sessions (>60 sec):", long_sessions[:5])

```

```

High ratings (>4): [False False False  True  True]
Long sessions (>60 sec): [False False False False False]

```

```

1 #Masks and Fancy Indexing
2 high_rating_users = users[ratings > 4]
3 long_sessions_values = durations[durations > 60]
4 print("Users with high ratings:", high_rating_users[:5])
5 print("Durations > 60 sec:", long_sessions_values[:5])

```

```

Users with high ratings: [ 500000 1700000 1700000 1700000 1700000]
Durations > 60 sec: []

```

```

1 #Fast Sorting with np.sort and np.argsort
2 sorted_ratings = np.sort(ratings)
3 print("Sorted ratings:", sorted_ratings[:10])
4
5 top_duration_indices = np.argsort(durations)[-5:]
6 print("Top 5 longest session durations:", durations[top_duration_indices])

```

```

Sorted ratings: [3 3 3 3 3 3 3 3 3 3]
Top 5 longest session durations: [60 60 60 60 60]

```

```

1 #Partial Sorting with np.partition
2 top5_fast = np.partition(durations, -5)[-5:]
3 print("Top 5 durations using partition:", top5_fast)

```

```

Top 5 durations using partition: [60 60 60 60 60]

```

```

1 import pandas as pd
2 import numpy as np
3
4 # Load CSV file
5 df = pd.read_csv('/content/deepseek_vs_chatgpt.csv')
6
7 # Extract columns as NumPy arrays

```

```
8 users = np.array(df['Active_Users'], dtype=np.int32)
9 ratings = np.array(df['User_Rating'], dtype=np.int32)
10 durations = np.array(df['Session_Duration_sec'], dtype=np.int32)
```

```
1 #Partial Sorting
2 top5_fast = np.partition(durations, -5)[-5:]
3 print("Top 5 session durations (unsorted):", top5_fast)
```

→ Top 5 session durations (unsorted): [60 60 60 60 60]

```
1 #Cumulative Sum
2 cumulative = np.cumsum(users[:5])
3 print("Cumulative user count:", cumulative)
```

→ Cumulative user count: [ 500000 1000000 1500000 2000000 3700000]