Mahatma Education Society's

Pillai College of Arts, Commerce & Science (Autonomous)

Affiliated to University of Mumbai

'NAAC Accredited 'A' grade (3 cycles)'
'Best College Award' by University of Mumbai
ISO 9001:2015 Certified

PROJECT REPORT ON

"Online Learning Behaviour"

IN PARTIAL FULFILLMENT OF

BACHELOR OF COMPUTER SCIENCE
SEMESTER 2025-26

SUBMITTED BY: Dhanashri Ananda Achare

ROLL NO:5254

Mahatma Education Society's

Pillai College of Arts, Commerce & Science

(Autonomous)

Affiliated to University of Mumbai

NAAC Accredited 'A' grade (3 cycles)

Best College Award by University of Mumbai

ISO 9001:2015 Certified

This is to verify that Miss.**Dhanashri Ananda Achare** of **BSC CS Semester IV** has completed the project work in the Subject of **Introduction of Data Science** during the academic year 2025-26 under the guidance of Prof.**Sanjana Bhangale** being the partial requirement for the fulfillment of the curriculum of **Bachelor of Computer Science University of Mumbai**

Place:

Date:

Name of the Signature of faculty Name &Signature of external

Title :Online Learning Behaviour

Dataset Description

The dataset provides information about 105 learners engaging in online education. It captures a wide range of variables including demographic data (Gender, Age), behavioral patterns (Study Hours, Number of Courses), and qualitative feedback (Platform satisfaction, Internet quality).

This project aims to explore the relationship between study habits and student satisfaction, identifying how different learning modes and internet access affect the overall learning experience.

The dataset contains 105 entries with 13 initial columns:

1. **Gender:** Male or Female (Categorical)
2. **Age:** Age brackets of learners (Categorical)
3. **Platform:** Primary platform used, e.g., YouTube, Udemy (Categorical)
4. **Subjects:** Area of study (Categorical)
5. **Mode:** Learning style, e.g., Visual, Auditory (Categorical)
6. **Internet:** Quality of internet access (Categorical)
7. **StudyHours:** Weekly time spent on study (Categorical/Numeric)
8. **Courses:** Number of courses enrolled (Categorical/Numeric)
9. **Rating:** Satisfaction level from 1 to 5 (Categorical/Numeric)
10. **Doubts:** Average doubts per week (Categorical/Numeric)
11. **Motivation:** Reason for learning (Categorical)
12. **DoubtMethod:** How doubts are resolved (Categorical)
13. **ObsDate:** Date of survey response (Date/Time)

Project Objectives & Methodology

The project was executed in a Jupyter Notebook using Python's pandas and numpy libraries. The analysis was divided into three levels:

1. Basic (Easy): Data cleaning, column standardization, and simple filtering of demographics.
2. Intermediate (Moderate): Converting textual categories (e.g., "5-10 hours") into numeric averages for mathematical analysis. Creating new metrics like the Study_Intensity_Score.
3. Advanced: Identifying correlations between study time and satisfaction, and performing time-based analysis on survey responses.

Data Analysis Questions:

- Data Loading & Inspection

1)Load the dataset into a Pandas DataFrame.

```
import pandas as pd
import numpy as np
# 1. Load the dataset
df = pd.read_csv('online_learning.csv')
df
```

	Gender	Age	Platform	Subjects	Mode	Internet	StudyHours\n	Courses\r\n	Rating\r\n	Doubts\r\n
0	Female	19-24	YouTube	Programming/Coding	Auditory	Good	21-30 hours	4-5 courses	4(Satisfied)	4-5 doubts
1	Male	19-24	YouTube	Programming/Coding	Auditory	Good	5-10 hours	1 course	3(Neutral)	4-5 doubts
2	Male	19-24	Other	Business/Marketing	Reading	Poor	More than 30 hours	2-3 courses	5(Very Satisfied)	6-10 doubts
3	Male	19-24	YouTube	Business/Marketing	NaN	Average	Less than 5 hours	1 course	NaN	1-2 doubt
4	Female	19-24	YouTube	Business/Marketing	Reading	Good	Less than 5 hours	1 course	1(Very Dissatisfied)	1-2 doubt

2)Display the first 5 rows of the dataset.

```
#Display first 5 rows
df.head()
```

	Gender	Age	Platform	Subjects	Mode	Internet	StudyHours\n	Courses\r\n	Rating\r\n	Doubts\r\n
0	Female	19-24	YouTube	Programming/Coding	Auditory	Good	21-30 hours	4-5 courses	4(Satisfied)	4-5 doubts
1	Male	19-24	YouTube	Programming/Coding	Auditory	Good	5-10 hours	1 course	3(Neutral)	4-5 doubts
2	Male	19-24	Other	Business/Marketing	Reading	Poor	More than 30 hours	2-3 courses	5(Very Satisfied)	6-10 doubts
3	Male	19-24	YouTube	Business/Marketing	NaN	Average	Less than 5 hours	1 course	NaN	1-2 doubt
4	Female	19-24	YouTube	Business/Marketing	Reading	Good	Less than 5 hours	1 course	1(Very Dissatisfied)	1-2 doubt

3)Display the total number of rows and columns.

```
#Total rows and columns
df.shape
```

```
(105, 13)
```

4) Print all column names.

```
#Print column names
df.columns
```

```
Index(['Gender', 'Age', 'Platform', 'Subjects', 'Mode', 'Internet',
       'StudyHours\n', 'Courses\r\n', 'Rating\r\n', 'Doubts\r\n',
       'Motivation\r\n', 'DoubtMethod\n', 'ObsDate\n'],
      dtype='object')
```

5)Check data types of each column.

```
#Check data types
df.dtypes
```

```
Gender            object
Age               object
Platform          object
Subjects          object
Mode              object
Internet          object
StudyHours\n      object
Courses\r\n       object
Rating\r\n        object
Doubts\r\n        object
Motivation\r\n    object
DoubtMethod\n     object
ObsDate\n         object
dtype: object
```

6)Identify columns with missing values.

```
#Identify missing values
df.isnull().sum()
```

```
Gender              7
Age                 6
Platform            6
Subjects            6
Mode                7
Internet            6
StudyHours\n        6
Courses\r\n         7
Rating\r\n          7
Doubts\r\n          6
Motivation\r\n      6
DoubtMethod\n       6
ObsDate\n           0
dtype: int64
```

- Basic Filtering

7) Filter all Female respondents.

```
#Basic Filtering
#Female respondents
df[df["Gender"] == "Female"]
```

	Gender	Age	Platform	Subjects	Mode	Internet	StudyHours\n	Courses\r\n	Rating\r\n	Doubts\r\n
0	Female	19-24	YouTube	Programming/Coding	Auditory	Good	21-30 hours	4-5 courses	4(Satisfied)	4-5 doubts
4	Female	19-24	YouTube	Business/Marketing	Reading	Good	Less than 5 hours	1 course	1(Very Dissatisfied)	1-2 doubt
5	Female	13-18	YouTube	Business/Marketing	Reading	Good	21-30 hours	6 or more courses	1(Very Dissatisfied)	4-5 doubts
8	Female	19-24	YouTube	Business/Marketing	Visual	Good	11-20 hours	4-5 courses	4(Satisfied)	4-5 doubts

8)List respondents in the 19–24 age group.

```
#Age group 19-24
df[df['Age'] == '19-24']
```

	Gender	Age	Platform	Subjects	Mode	Internet	Stud
0	Female	19-24	YouTube	Programming/Coding	Auditory	Good	21
1	Male	19-24	YouTube	Programming/Coding	Auditory	Good	5
2	Male	19-24	Other	Business/Marketing	Reading	Poor	Mo
3	Male	19-24	YouTube	Business/Marketing	NaN	Average	Le
4	Female	19-24	YouTube	Business/Marketing	Reading	Good	Le

9)Show respondents who prefer YouTube as their learning platform.

```
#Prefer YouTube
df[df["Platform"] == "YouTube"]
```

	Gender	Age	Platform	Subjects	Mode	Internet	Stu
0	Female	19-24	YouTube	Programming/Coding	Auditory	Good	2
1	Male	19-24	YouTube	Programming/Coding	Auditory	Good	
3	Male	19-24	YouTube	Business/Marketing	NaN	Average	
4	Female	19-24	YouTube	Business/Marketing	Reading	Good	
5	Female	13-18	YouTube	Business/Marketing	Reading	Good	2

10)Filter respondents who spend More than 30 hours on online study per week.

```
#More than 30 study hours
df[df["StudyHours\n"] == "More than 30 hours"]
```

	Gender	Age	Platform	Subjects	Mode	Internet	StudyHours\n
2	Male	19-24	Other	Business/Marketing	Reading	Poor	More than 30 hours
12	Female	13-18	Other	Other	Kinesthetic	Good	More than 30 hours
27	Male	19-24	Coursera	STEM(Science,Technology,Engineering,Mathematics)	Reading	Average	More than 30 hours
47	Male	19-24	YouTube	Business/Marketing	Visual	Good	More than 30 hours

11) List learners who are enrolled in only 1 course.

```
#Only 1 course
df[df["Courses\r\n"] == "1 course"]
```

	Gender	Age	Platform	Subjects	Mode	Internet	StudyHours\n	Courses\r\n
1	Male	19-24	YouTube	Programming/Coding	Auditory	Good	5-10 hours	1 course
3	Male	19-24	YouTube	Business/Marketing	NaN	Average	Less than 5 hours	1 course
4	Female	19-24	YouTube	Business/Marketing	Reading	Good	Less than 5 hours	1 course
6	Male	19-24	Other	Business/Marketing	Visual	Good	11-20 hours	1 course

- Frequency & Counts

12)Count how many respondents prefer each online learning platform.

```
#Frequency & Counts
#Platform preference count
df["Platform"].value_counts()
```

```
Platform
YouTube     59
Other       18
Coursera    11
Udemy       11
Name: count, dtype: int64
```

13)Find the most common subject preference.

```
#Most common subject
df["Subjects"].mode()
```

```
0    Programming/Coding
Name: Subjects, dtype: object
```

14)Count the number of respondents by type of learner (Visual, Auditory, Reading, Kinesthetic).

```
#Learner type count
df["Mode"].value_counts()
```

```
Mode
Reading        42
Visual         37
Auditory       14
Kinesthetic     5
Name: count, dtype: int64
```

15)Count how many respondents rated their internet access as Good.

```
#Good internet access count
df[df["Internet"] == "Good"].shape[0]
```

```
67
```

- MODERATE LEVEL QUESTIONS
  - ❖ Text & Numeric Cleaning

16) Convert hours spent on online study per week into numeric values:

• Less than 5 hours

• 5–10 hours

• 11–20 hours

• 21–30 hours

• More than 30 hours

```
#MODERATE LEVEL
#Text & Numeric Cleaning
#Convert study hours to numeric
hours_map = {
    "Less than 5 hours": 3,
    "5-10 hours": 8,
    "11-20 hours": 15,
    "21-30 hours": 25,
    "More than 30 hours": 35
}

df["StudyHours\n_Num"] = df["StudyHours\n"].map(hours_map)
print(df["StudyHours\n_Num"])
```

```
0       NaN
1       NaN
2      35.0
3       3.0
4       3.0
       ...
100     NaN
101     3.0
102     NaN
103     3.0
104     NaN
Name: StudyHours\n_Num, Length: 105, dtype: float64
```

17)Convert number of online courses enrolled into numeric values:

• 1 course

• 2–3 courses

• 4–5 courses

• 6 or more courses

```python
#Convert courses to numeric
courses_map = {
    "1 course": 1,
    "2-3 courses": 2.5,
    "4-5 courses": 4.5,
    "6 or more courses": 6
}

df["Courses\r\n_Num"] = df["Courses\r\n"].map(courses_map)
print(df["Courses\r\n_Num"])
```

```
      ...
100    1.0
101    1.0
102    1.0
103    NaN
104    NaN
Name: Courses\r\n_Num, Length: 105, dtype: float64
```

18)Extract numeric rating from:

- 4(Satisfied)
- 3(Neutral)
- 1(Very Dissatisfied)

```python
#Extract numeric rating
df["Rating\r\n_Num"] = df["Rating\r\n"].str.extract('(\d)').astype(float)
print(df["Rating\r\n_Num"])
```

```
0      4.0
1      3.0
2      5.0
3      NaN
4      1.0
      ...
100    3.0
101    4.0
102    4.0
103    3.0
104    4.0
Name: Rating\r\n_Num, Length: 105, dtype: float64
```

19)Convert number of doubts per week into numeric averages:

• 1–2 doubt

• 4–5 doubts

• More than 10 doubts

```
#Convert doubts into numeric
doubt_map = {
    "1-2 doubt": 1.5,
    "4-5 doubts": 4.5,
    "More than 10 doubts": 10
}

df["Doubts\r\n_Num"] = df["Doubts\r\n"].map(doubt_map)
print(df["Doubts\r\n_Num"])
```

```
       ..
100   NaN
101   NaN
102   NaN
103   NaN
104   NaN
Name: Doubts\r\n_Num, Length: 105, dtype: float64
```

20) Standardize inconsistent text values (e.g., doubt vs doubts, missing learner types).

```
#Standardize text values
df["Doubts\r\n"] = df["Doubts\r\n"].str.replace("doubt", "doubts")
df["Mode"] = df["Mode"].fillna("Unknown")
print(df)
```

```
        Doubts\r\n                        Motivation\r\n  \
0     4-5 doubtsss       To improve academic performance
1     4-5 doubtsss  Because its more flexible/convenient
2    6-10 doubtsss       To improve academic performance
3      1-2 doubtss                    To gain new skills
4      1-2 doubtss                    To gain new skills
..             ...                                   ...
100    1-2 doubtss                    To gain new skills
101   2-3 doubtsss  Because its more flexible/convenient
102   4-5 doubtsss                    To gain new skills
103   4-5 doubtsss       To improve academic performance
104   4-5 doubtsss       To improve academic performance
```

- Feature Engineering

21)Create a new column:Study_Intensity_Score= weekly_study_hours × number_of_courses

```
#Feature Engineering
#Study Intensity Score
df["Study_Intensity_Score"] = df["StudyHours\n_Num"] * df["Courses\r\n_Num"]
print(df)
```

```
     Courses\r\n_Num  Rating\r\n_Num  Doubts\r\n_Num  Study_Intensity_Score
0                NaN             4.0             NaN                    NaN
1                1.0             3.0             NaN                    NaN
2                NaN             5.0             NaN                    NaN
3                1.0             NaN             NaN                    3.0
4                1.0             1.0             NaN                    3.0
```

22)Create a binary column:High_Engagement

• Yes → study hours ≥ 21 and courses ≥ 4 • No → otherwise

```
#High Engagement column
df["High_Engagement"] = df.apply(
    lambda x: "Yes" if x["StudyHours\n_Num"] >= 21 and x["Courses\r\n_Num"] >= 4 else "No",
    axis=1
)
print(df)
```

```
     High_Engagement
0                 No
1                 No
2                 No
3                 No
4                 No
```

23)Create Satisfaction_Level categories:• Low (1–2) • Medium (3) • High (4–5)

```
#Satisfaction Level
def satisfaction_level(x):
    if x <= 2:
        return "Low"
    elif x == 3:
        return "Medium"
    else:
        return "High"

df["Satisfaction_Level"] = df["Rating\r\n_Num"].apply(satisfaction_level)
print(df)
```

	High_Engagement	Satisfaction_Level
0	No	High
1	No	Medium
2	No	High
3	No	High
4	No	Low

24)Create a column classifying Internet Quality as numeric:• Poor → 1

• Average → 2 • Good → 3

```
#Internet quality numeric
internet_map = {
    "Poor": 1,
    "Average": 2,
    "Good": 3
}

df["Internet_Num"] = df["Internet"].map(internet_map)
print(df)
```

	Study_Intensity_Score	High_Engagement	Satisfaction_Level	Internet_Num
0	NaN	No	High	3.0
1	NaN	No	Medium	3.0
2	NaN	No	High	1.0
3	3.0	No	High	2.0
4	3.0	No	Low	3.0

25)Create a column Active_Learner:• Yes → doubts ≥ 4 per week • No → otherwise

```
#Active learner column
df["Active_Learner"] = df["Doubts\r\n_Num"].apply(lambda x: "Yes" if x >= 4 else "No")
print(df)
```

	Satisfaction_Level	Internet_Num	Active_Learner
0	High	3.0	No
1	Medium	3.0	No
2	High	1.0	No
3	High	2.0	No
4	Low	3.0	No

- Grouping & Aggregation

26)Find average weekly study hours by gender.

```
#Grouping & Aggregation
#Average study hours by gender
df.groupby("Gender")["StudyHours\n_Num"].mean()
```

```
Gender
Female     5.909091
Male      10.619048
Name: StudyHours\n_Num, dtype: float64
```

27) Find average course rating by learning platform.

```
#Average rating by platform
df.groupby("Platform")["Rating\r\n_Num"].mean()
```

```
Platform
Coursera    2.454545
Other       2.888889
Udemy       3.363636
YouTube     3.258621
Name: Rating\r\n_Num, dtype: float64
```

28)Group by subject preference and count respondents.

```
#Respondents by subject
df["Subjects"].value_counts()
```

```
Subjects
Programming/Coding                                  33
Business/Marketing                                  25
Other                                               19
STEM(Science,Technology,Engineering,Mathematics)    15
Humanities(Literature,History,Philosophy)            7
Name: count, dtype: int64
```

29)Find average number of doubts per week by type of learner.

```
#Average doubts by learner type
df.groupby("Mode")["Doubts\r\n_Num"].mean()
```

```
Mode
Auditory       10.0
Kinesthetic    10.0
Reading        10.0
Unknown         NaN
Visual         10.0
Name: Doubts\r\n_Num, dtype: float64
```

30)Group by age group and calculate average study hours.

```
#Average study hours by age group
df.groupby("Age")["StudyHours\n_Num"].mean()
```

```
Age
13-18    12.142857
19-24     7.444444
25-30     3.000000
Name: StudyHours\n Num, dtype: float64
```

- ADVANCED–MODERATE QUESTIONS

❖ Time-Based Analysis

31) Convert Observation_Date into datetime format.

```
#ADVANCED-MODERATE
#Time Based Analysis
#Convert date to datetime
df["ObsDate\n"] = pd.to_datetime(df["ObsDate\n"])
print(df)
```

```
       Doubts\r\n  ...  ObsDate\n StudyHours\n_Num Courses\r\n
0     4-5 doubtsss  ... 2025-01-01              NaN
1     4-5 doubtsss  ... 2025-01-02              NaN
2    6-10 doubtsss  ... 2025-01-03             35.0
3      1-2 doubtss  ... 2025-01-04              3.0
4      1-2 doubtss  ... 2025-01-05              3.0
..             ...  ...        ...              ...
```

32)Find number of survey responses collected per day.

```
#Responses per day
df["ObsDate\n"].value_counts()
```

```
2025-02-02    1
2025-02-01    1
2025-01-31    1
2025-01-30    1
2025-04-15    1
Name: count, Length: 105, dtype: int64
```

33)Identify the date with the highest number of high engagement learners.

```
#Date with highest high engagement
df[df["High_Engagement"] == "Yes"]["ObsDate\n"].value_counts().idxmax()
```

```
Timestamp('2025-04-07 00:00:00')
```

Correlation & Insights

34)Check correlation between:

• Weekly study hours and course rating

• Number of doubts and satisfaction rating

• Number of courses and study hours

```
#Correlation & Insights
#Correlation
df[["StudyHours\n_Num","Rating\r\n_Num","Doubts\r\n_Num","Courses\r\n_Num"]].corr()
```

	StudyHours\n_Num	Rating\r\n_Num	Doubts\r\n_Num	Courses\r\n_Num
StudyHours\n_Num	1.000000	-0.009430	NaN	0.266667
Rating\r\n_Num	-0.009430	1.000000	NaN	-0.007762
Doubts\r\n_Num	NaN	NaN	NaN	NaN
Courses\r\n_Num	0.266667	-0.007762	NaN	1.000000

35) Does better internet access lead to higher satisfaction?

```
#Internet vs Satisfaction
df.groupby("Internet")["Rating\r\n_Num"].mean()
```

```
Internet
Average    3.464286
Good       2.970149
Poor       3.000000
Name: Rating\r\n_Num, dtype: float64
```

Conclusion

This analysis demonstrates that modern online learning is highly diverse. While platforms like YouTube dominate the landscape, the "one-size-fits-all" approach to study hours does not directly translate to student satisfaction. Factors such as **Internet Quality** and **Doubt Resolution Methods** play a more significant role in the learner's journey than simply the volume of study time.

The project highlights the importance of **Feature Engineering** in data science—turning text-based survey answers into numeric scores allowed us to uncover hidden trends that weren't visible in the raw CSV file.