

```
In [1]: 1 import numpy as np
        2 import pandas as pd
        3 import seaborn as sns
        4 import matplotlib.pyplot as plt
```

```
In [2]: 1 pd.options.display.max_columns = None
```

```
In [3]: 1 df_orignal = pd.read_csv('Time-Wasters on Social Media.csv')
```

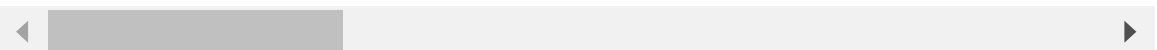
```
In [4]: 1 df = df_orignal.copy()
```

```
In [5]: 1 df
```

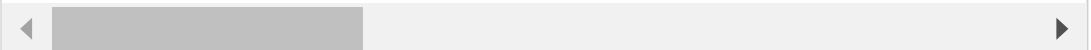
Out[5]:

	UserID	Age	Gender	Location	Income	Debt	Owns Property	Profession	Demographics	
0	1	56	Male	Pakistan	82812	True	True	Engineer	Rural	Ir
1	2	46	Female	Mexico	27999	False	True	Artist	Urban	Ir
2	3	32	Female	United States	42436	False	True	Engineer	Rural	F
3	4	60	Male	Barzil	62963	True	False	Waiting staff	Rural	
4	5	25	Male	Pakistan	22096	False	True	Manager	Urban	
...	
995	996	22	Male	India	74254	True	True	Students	Rural	
996	997	40	Female	Pakistan	27006	False	False	Waiting staff	Urban	F
997	998	27	Male	India	94218	True	True	Waiting staff	Rural	
998	999	61	Male	Pakistan	85344	True	False	Students	Urban	
999	1000	19	Male	India	53840	True	True	driver	Urban	

1000 rows × 31 columns



```
In [6]: 1 df.drop(columns=['Demographics', 'Income', 'Profession', 'CurrentActivity'])
```



```
In [7]: 1 df.drop(columns=['Video Length', 'Time Spent On Video'], inplace=True)
```

```
In [8]: 1 df
```

```
Out[8]:
```

	Age	Gender	Location	Debt	Owns Property	Platform	Total Time Spent	Video Category	Engagement	Frec
0	56	Male	Pakistan	True	True	Instagram	80	Pranks	7867	
1	46	Female	Mexico	False	True	Instagram	228	Pranks	5944	Aft
2	32	Female	United States	False	True	Facebook	30	Vlogs	8674	E
3	60	Male	Barzil	True	False	YouTube	101	Vlogs	2477	
4	25	Male	Pakistan	False	True	TikTok	136	Gaming	3093	M
...
995	22	Male	India	True	True	TikTok	144	Gaming	5179	Aft
996	40	Female	Pakistan	False	False	Facebook	231	Life Hacks	1803	
997	27	Male	India	True	True	TikTok	165	Pranks	9290	
998	61	Male	Pakistan	True	False	YouTube	151	Life Hacks	2050	
999	19	Male	India	True	True	YouTube	167	Pranks	9790	E

1000 rows × 16 columns



```
In [9]: 1 temp_df = df.copy()
```

```
In [10]: 1 df['Addiction Level'].value_counts()
```

```
Out[10]: Addiction Level
2      248
5      228
0      180
3      159
1       60
7       55
4       36
6       34
Name: count, dtype: int64
```

```
In [11]: 1 df['Addiction Level'] = df['Addiction Level'].replace({0: 'no addictio
2                                     2: 'low', 3: 'mod
3                                     5: 'high', 6: 'hi
```

```
In [12]: 1 df['Addiction Level'].value_counts()
```

```
Out[12]: Addiction Level
low          308
high         262
moderate     195
no addiction  180
extreme       55
Name: count, dtype: int64
```

```
In [13]: 1 df['Total Time Spent'].describe()
```

```
Out[13]: count    1000.000000
mean       151.406000
std        83.952637
min         10.000000
25%        78.000000
50%       152.000000
75%       223.000000
max       298.000000
Name: Total Time Spent, dtype: float64
```

```
In [14]: 1 temp_df = temp_df.select_dtypes(exclude=['object', 'bool'])
```

```
In [15]: 1 temp_df.corr()['Addiction Level']
```

```
Out[15]: Age          0.033493
Total Time Spent     0.016086
Engagement           0.027620
ProductivityLoss    -0.994939
Satisfaction         0.994939
Self Control        -1.000000
Addiction Level      1.000000
Name: Addiction Level, dtype: float64
```

In [16]:

1 df

Out[16]:

	Age	Gender	Location	Debt	Owns Property	Platform	Total Time Spent	Video Category	Engagement	Frec
0	56	Male	Pakistan	True	True	Instagram	80	Pranks	7867	
1	46	Female	Mexico	False	True	Instagram	228	Pranks	5944	Aft
2	32	Female	United States	False	True	Facebook	30	Vlogs	8674	E
3	60	Male	Barzil	True	False	YouTube	101	Vlogs	2477	
4	25	Male	Pakistan	False	True	TikTok	136	Gaming	3093	M
...
995	22	Male	India	True	True	TikTok	144	Gaming	5179	Aft
996	40	Female	Pakistan	False	False	Facebook	231	Life Hacks	1803	
997	27	Male	India	True	True	TikTok	165	Pranks	9290	
998	61	Male	Pakistan	True	False	YouTube	151	Life Hacks	2050	
999	19	Male	India	True	True	YouTube	167	Pranks	9790	E

1000 rows × 16 columns



In [17]:

```
1 lower = df['Total Time Spent'].quantile(.33)
2 medium = df['Total Time Spent'].quantile(.66)
3 higher = df['Total Time Spent'].quantile(1)
```

In [18]:

1 higher, lower, medium

Out[18]: (298.0, 101.0, 197.0)

In [19]:

```
1 def convert(time):
2     if time<=101:
3         return 'less'
4     elif 101<time<=197:
5         return 'moderate'
6     elif 197<time<=298:
7         return 'high'
8     else:
9         return None
10
```

In [20]: 1 df['Total Time Spent'] = df['Total Time Spent'].apply(convert)

In [21]: 1 df['Total Time Spent'].value_counts()

Out[21]: Total Time Spent
 high 336
 less 334
 moderate 330
 Name: count, dtype: int64

In [22]: 1 df

Out[22]:

	Age	Gender	Location	Debt	Owns Property	Platform	Total Time Spent	Video Category	Engagement	F
0	56	Male	Pakistan	True	True	Instagram	less	Pranks	7867	
1	46	Female	Mexico	False	True	Instagram	high	Pranks	5944	
2	32	Female	United States	False	True	Facebook	less	Vlogs	8674	
3	60	Male	Barzil	True	False	YouTube	less	Vlogs	2477	
4	25	Male	Pakistan	False	True	TikTok	moderate	Gaming	3093	
...	
995	22	Male	India	True	True	TikTok	moderate	Gaming	5179	
996	40	Female	Pakistan	False	False	Facebook	high	Life Hacks	1803	
997	27	Male	India	True	True	TikTok	moderate	Pranks	9290	
998	61	Male	Pakistan	True	False	YouTube	moderate	Life Hacks	2050	
999	19	Male	India	True	True	YouTube	moderate	Pranks	9790	

1000 rows × 16 columns



In [23]: 1 df['Engagement'].describe()

Out[23]: count 1000.000000
 mean 4997.159000
 std 2910.053701
 min 15.000000
 25% 2415.750000
 50% 5016.000000
 75% 7540.250000
 max 9982.000000
 Name: Engagement, dtype: float64

```
In [24]: 1 def convert1(num):
2         if num<=3000:
3             return 'less'
4         elif 3000<num<=6000:
5             return 'moderate'
6         elif 6000<num<=9982:
7             return 'high'
8         else:
9             return None
```

```
In [25]: 1 df['Engagement'] = df['Engagement'].apply(convert1)
```

```
In [26]: 1 df['Engagement'].value_counts()
```

```
Out[26]: Engagement
high      403
less      306
moderate  291
Name: count, dtype: int64
```

```
In [27]: 1 df.to_csv("test.csv",index=False)
2
```

```
In [28]: 1 from sklearn.preprocessing import OrdinalEncoder
2
3 # Create a copy of the original data for label encoding
4 data_label_encoded = df.copy()
5
6 categorical_cols = df.select_dtypes(include=['object','bool']).columns
7
8 # Apply label encoding to categorical columns
9 for col in categorical_cols:
10     oe = OrdinalEncoder()
11     data_label_encoded[col] = oe.fit_transform(data_label_encoded[[col]
12
13 # Splitting the dataset into training and testing sets
14 X_label = data_label_encoded.drop('Addiction Level', axis=1)
15 y_label = data_label_encoded['Addiction Level']
```

In [29]:

1 X_label

Out[29]:

	Age	Gender	Location	Debt	Owns Property	Platform	Total Time Spent	Video Category	Engagement	Frequ
0	56	1.0	6.0	1.0	1.0	1.0	1.0	6.0	0.0	
1	46	0.0	5.0	0.0	1.0	1.0	0.0	6.0	2.0	
2	32	0.0	8.0	0.0	1.0	0.0	1.0	8.0	0.0	
3	60	1.0	0.0	1.0	0.0	3.0	1.0	8.0	1.0	
4	25	1.0	6.0	0.0	1.0	2.0	2.0	3.0	2.0	
...
995	22	1.0	2.0	1.0	1.0	2.0	2.0	3.0	2.0	
996	40	0.0	6.0	0.0	0.0	0.0	0.0	5.0	1.0	
997	27	1.0	2.0	1.0	1.0	2.0	2.0	6.0	0.0	
998	61	1.0	6.0	1.0	0.0	3.0	2.0	5.0	1.0	
999	19	1.0	2.0	1.0	1.0	3.0	2.0	6.0	0.0	

1000 rows × 15 columns

In [47]:

1 df.to_csv('df.csv',index=False)

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1