# PythonMiniProject_Kirti_Gupta (1)

August 11, 2025

## 1 Python mini project on Social Media Addiction by Kirti Gupta

[42]:
```
# Importing libraries needed for the project

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

### 1.0.1 Understanding the data

[43]:
```
# Loading the dataset into a dataframe

df = pd.read_csv('/content/Students Social Media Addiction (1).csv')
```

[44]:
```
# Showing the first five records from the dataframe using head()

df.head()
```

[44]:

| | Student_ID | Age | Gender | Academic_Level | Country | Avg_Daily_Usage_Hours | Most_Used_Platform | Affects_Academic_Performance | Sleep_Hours_Per_Night | Mental_Health_Score | Relationship_Status | Conflicts_Over_Social_Media |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 19 | Female | Undergraduate | Bangladesh | 5.2 | Instagram | Yes | 6.5 | | | |
| 1 | 2 | 22 | Male | Graduate | India | 2.1 | Twitter | No | 7.5 | | | |
| 2 | 3 | 20 | Female | Undergraduate | USA | 6.0 | TikTok | Yes | 5.0 | | | |
| 3 | 4 | 18 | Male | High School | UK | 3.0 | YouTube | No | 7.0 | | | |
| 4 | 5 | 21 | Male | Graduate | Canada | 4.5 | Facebook | Yes | 6.0 | | | |

```
0                    6     In Relationship                           3
1                    8              Single                           0
2                    5         Complicated                           4
3                    7              Single                           1
4                    6     In Relationship                           2

   Addicted_Score
0               8
1               3
2               9
3               4
4               7
```

[45]:
```
# Showing a random sample of 5 records using sample()

df.sample(5)
```

[45]:
```
     Student_ID  Age  Gender Academic_Level     Country  \
184         185   19  Female  Undergraduate  Bangladesh
32           33   18    Male    High School   Indonesia
74           75   23    Male       Graduate     Belarus
604         605   22  Female       Graduate       Spain
385         386   21    Male       Graduate      Turkey

     Avg_Daily_Usage_Hours Most_Used_Platform Affects_Academic_Performance  \
184                    4.6          Instagram                          Yes
32                     5.4             TikTok                          Yes
74                     2.5           LinkedIn                           No
604                    6.3             TikTok                          Yes
385                    4.9          Instagram                          Yes

     Sleep_Hours_Per_Night  Mental_Health_Score Relationship_Status  \
184                    7.0                    5              Single
32                     5.4                    5         Complicated
74                     7.3                    8     In Relationship
604                    6.3                    5              Single
385                    7.3                    6              Single

     Conflicts_Over_Social_Media  Addicted_Score
184                            3               7
32                             4               8
74                             1               4
604                            4               8
385                            3               7
```

[46]:
```
# Check the number of columns and rows in the dataframe
```

```
print('Number of rows:', df.shape[0])
print('Number of columns:', df.shape[1])
```

```
Number of rows: 705
Number of columns: 13
```

[47]:
```
# Showing Five point summary, count, mean & standard deviation using describe()

df.describe()
```

[47]:

| | Student_ID | Age | Avg_Daily_Usage_Hours | Sleep_Hours_Per_Night | \ |
|---|---|---|---|---|---|
| count | 705.000000 | 705.000000 | 705.000000 | 705.000000 | |
| mean | 353.000000 | 20.659574 | 4.918723 | 6.868936 | |
| std | 203.660256 | 1.399217 | 1.257395 | 1.126848 | |
| min | 1.000000 | 18.000000 | 1.500000 | 3.800000 | |
| 25% | 177.000000 | 19.000000 | 4.100000 | 6.000000 | |
| 50% | 353.000000 | 21.000000 | 4.800000 | 6.900000 | |
| 75% | 529.000000 | 22.000000 | 5.800000 | 7.700000 | |
| max | 705.000000 | 24.000000 | 8.500000 | 9.600000 | |

| | Mental_Health_Score | Conflicts_Over_Social_Media | Addicted_Score |
|---|---|---|---|
| count | 705.000000 | 705.000000 | 705.000000 |
| mean | 6.226950 | 2.849645 | 6.436879 |
| std | 1.105055 | 0.957968 | 1.587165 |
| min | 4.000000 | 0.000000 | 2.000000 |
| 25% | 5.000000 | 2.000000 | 5.000000 |
| 50% | 6.000000 | 3.000000 | 7.000000 |
| 75% | 7.000000 | 4.000000 | 8.000000 |
| max | 9.000000 | 5.000000 | 9.000000 |

### 1.0.2 Data cleaning

[48]:
```
# check missing values

df.isnull().sum()
```

[48]:

| | |
|---|---|
| Student_ID | 0 |
| Age | 0 |
| Gender | 0 |
| Academic_Level | 0 |
| Country | 0 |
| Avg_Daily_Usage_Hours | 0 |
| Most_Used_Platform | 0 |
| Affects_Academic_Performance | 0 |
| Sleep_Hours_Per_Night | 0 |
| Mental_Health_Score | 0 |
| Relationship_Status | 0 |

```
Conflicts_Over_Social_Media     0
Addicted_Score                  0
dtype: int64
```

- *From the above code we found that there are no null values*
- *But if there were any null values then to handle missing values following will be applied*

**We can use two methods for cleaning null values**

1. Dropping the null value rows

[49]:
```
df.dropna(inplace = True)
```

2. Filling missing values in numerical columns with the mean of the column

[50]:
```
# Checking missing values before applying fillna
print("Missing values BEFORE applying fillna:")
print(df.isnull().sum())
print('-' * 50)

numerical_cols = df.select_dtypes(include = np.number).columns

for col in numerical_cols:
    df[col].fillna(df[col].mean(), inplace = True)

# Checking missing values after applying fillna
print("\nMissing values AFTER applying fillna:")
print(df.isnull().sum())
```

```
Missing values BEFORE applying fillna:
Student_ID                      0
Age                             0
Gender                          0
Academic_Level                  0
Country                         0
Avg_Daily_Usage_Hours           0
Most_Used_Platform              0
Affects_Academic_Performance    0
Sleep_Hours_Per_Night           0
Mental_Health_Score             0
Relationship_Status             0
Conflicts_Over_Social_Media     0
Addicted_Score                  0
dtype: int64
--------------------------------------------------

Missing values AFTER applying fillna:
Student_ID                      0
Age                             0
```

```
Gender                          0
Academic_Level                  0
Country                         0
Avg_Daily_Usage_Hours           0
Most_Used_Platform              0
Affects_Academic_Performance    0
Sleep_Hours_Per_Night           0
Mental_Health_Score             0
Relationship_Status             0
Conflicts_Over_Social_Media     0
Addicted_Score                  0
dtype: int64
```

[51]:
```
# Identify incorrect datatypes

df.dtypes
```

[51]:
```
Student_ID                       int64
Age                              int64
Gender                          object
Academic_Level                  object
Country                         object
Avg_Daily_Usage_Hours          float64
Most_Used_Platform              object
Affects_Academic_Performance    object
Sleep_Hours_Per_Night          float64
Mental_Health_Score              int64
Relationship_Status             object
Conflicts_Over_Social_Media      int64
Addicted_Score                   int64
dtype: object
```

- *As seen on the above result datatypes are good and does not require any cleaning*
- *But if there were any dataypes mismatch then we can use the following*

**We can use two methods for datatype correction**

1. Using astype()

[52]:
```
# lets assume Age column (integer) has incorrect datatype (ex: string),
# the following code will convert it to int

df['Age'] = df['Age'].astype(int)
```

2. using pd.to_numeric()

[53]:
```
# Check the datatype before conversion
print("\nDatatype of 'Avg_Daily_Usage_Hours' BEFORE conversion:")
print(df['Avg_Daily_Usage_Hours'].dtype)
```

```
print('-' * 80)

# Convert 'Avg_Daily_Usage_Hours' to numeric
df['Avg_Daily_Usage_Hours'] = pd.to_numeric(df['Avg_Daily_Usage_Hours'], errors␣
 ↪= 'coerce')

# Check for any values that couldn't be converted
print("\nNumber of non-numeric values after converting 'Avg_Daily_Usage_Hours':
 ↪")
print(df['Avg_Daily_Usage_Hours'].isnull().sum())
print('-' * 80)

# Verify the datatype
print("\nDatatype of 'Avg_Daily_Usage_Hours' AFTER conversion:")
print(df['Avg_Daily_Usage_Hours'].dtype)
```

```
Datatype of 'Avg_Daily_Usage_Hours' BEFORE conversion:
float64
--------------------------------------------------------------------------------

Number of non-numeric values after converting 'Avg_Daily_Usage_Hours':
0
--------------------------------------------------------------------------------

Datatype of 'Avg_Daily_Usage_Hours' AFTER conversion:
float64
```

### 1.0.3 Understanding relationships between Age, Gender & Daily Usage

[54]:
```
# Average daily usage hours by age
usage_by_age = df.groupby('Age')['Avg_Daily_Usage_Hours'].mean()
print("Average Daily Usage by Age:")
print(f"{usage_by_age}\n")
print('-' * 50)

# Average daily usage hours by gender
print("\nAverage Daily Usage by Gender:")
usage_by_gender = df.groupby('Gender')['Avg_Daily_Usage_Hours'].mean()
print(usage_by_gender)
```

```
Average Daily Usage by Age:
Age
18    5.385714
19    5.120245
20    4.930303
21    4.950641
```

```
22    4.676190
23    4.508824
24    5.046154
Name: Avg_Daily_Usage_Hours, dtype: float64

--------------------------------------------------

Average Daily Usage by Gender:
Gender
Female    5.011048
Male      4.826136
Name: Avg_Daily_Usage_Hours, dtype: float64
```

**Age, Gender & Daily Usage data observations :** - *Students aged 18 have the highest average daily usage and students aged 23 have the lowest.* - *The average daily social media usage is slightly higher for female students.*

### 1.0.4 Understanding relationships between Sleep patterns, Academic performance & Social interaction

[55]:
```
# Average sleep hours per night by affect on affect on academic performance
print("Relationship between Sleep Hours and Academic Performance:")
academic_vs_sleepHours = df.
 ↪groupby('Affects_Academic_Performance')['Sleep_Hours_Per_Night'].mean()
print(f"{academic_vs_sleepHours}\n")
print('-' * 50)

# Average mental health score by sleep hours per night
print("\nRelationship between Sleep Hours and Mental Health Score:")
sleepHours_vs_mentalHealth = df.
 ↪groupby('Sleep_Hours_Per_Night')['Mental_Health_Score'].mean()
print(f"{sleepHours_vs_mentalHealth.head(10)}\n")
print('-' * 50)

# Average social media conflicts by relationship status
print("\nRelationship between Conflicts over Social Media and Relationship␣
 ↪Status:")
rel_status_vs_conflict = df.
 ↪groupby('Relationship_Status')['Conflicts_Over_Social_Media'].mean()
print(rel_status_vs_conflict)
```

```
Relationship between Sleep Hours and Academic Performance:
Affects_Academic_Performance
No     7.813095
Yes    6.343709
Name: Sleep_Hours_Per_Night, dtype: float64

--------------------------------------------------
```

```
Relationship between Sleep Hours and Mental Health Score:
Sleep_Hours_Per_Night
3.8    5.000000
3.9    5.000000
4.0    5.000000
4.1    5.500000
4.2    5.500000
4.3    5.500000
4.4    5.500000
4.5    5.000000
4.6    5.666667
4.7    5.666667
Name: Mental_Health_Score, dtype: float64

--------------------------------------------------

Relationship between Conflicts over Social Media and Relationship Status:
Relationship_Status
Complicated        3.031250
In Relationship    2.761246
Single             2.901042
Name: Conflicts_Over_Social_Media, dtype: float64
```

**Sleep patterns, Academic performance & Social interaction data observations :** - *Students whose academic performance is affected by social media tend to get less sleep.* - *There seems to be a trend that students who get fewer hours of sleep per night tend to have lower mental health scores.* - *Students in 'Complicated' relationships report slightly more conflicts over social media compared to those who are 'Single' or 'In Relationship'.*

### 1.0.5 Addiction variation across demographics

[56]:
```python
# Average addiction score by Gender
print("Average Addiction Score by Gender:")
print(f"{df.groupby('Gender')['Addicted_Score'].mean()}\n")
print('-' * 50)

# Average addiction score by Academic Level
print("\nAverage Addiction Score by Academic Level:")
print(f"{df.groupby('Academic_Level')['Addicted_Score'].mean()}\n")
print('-' * 50)

# Average addiction score by Age
print("\nAverage Addiction Score by Age:")
print(df.groupby('Age')['Addicted_Score'].mean())
```

```
Average Addiction Score by Gender:
Gender
```

```
Female    6.515581
Male      6.357955
Name: Addicted_Score, dtype: float64

------------------------------------------------

Average Addiction Score by Academic Level:
Academic_Level
Graduate         6.243077
High School      8.037037
Undergraduate    6.492918
Name: Addicted_Score, dtype: float64

------------------------------------------------

Average Addiction Score by Age:
Age
18    7.785714
19    6.650307
20    6.478788
21    6.589744
22    6.095238
23    5.676471
24    6.115385
Name: Addicted_Score, dtype: float64
```

**Addiction across demographic (gender, academic level & age) observations :** - *Addiction score tends to be higher in Females but only by a slight margin.* - *High School students tend to have higher addiction score.* - *Students aged 18 years have the highest addiction score.*

### 1.0.6 Average addiction level across different Genders

[57]:
```
print("Average addiction level by Gender:")
print('-' * 40)
print(df.groupby('Gender')['Addicted_Score'].mean())
```

```
Average addiction level by Gender:
----------------------------------------
Gender
Female    6.515581
Male      6.357955
Name: Addicted_Score, dtype: float64
```

### 1.0.7 Average addiction level across different Age groups

[58]:
```
print("\nAverage addiction level by Age group:")
print('-' * 40)
print(df.groupby('Age')['Addicted_Score'].mean())
```

```
Average addiction level by Age group:
----------------------------------------
Age
18    7.785714
19    6.650307
20    6.478788
21    6.589744
22    6.095238
23    5.676471
24    6.115385
Name: Addicted_Score, dtype: float64
```

### 1.0.8 Average addiction level across different Education levels

[59]:
```
print("\nAverage addiction level by Education level:")
print('-' * 45)
print(df.groupby('Academic_Level')['Addicted_Score'].mean())
```

```
Average addiction level by Education level:
---------------------------------------------
Academic_Level
Graduate         6.243077
High School      8.037037
Undergraduate    6.492918
Name: Addicted_Score, dtype: float64
```

### 1.0.9 Classifying risk level (Low/Medium/High) based on usage hours

[60]:
```
def classify_risk(usage_hours):
    '''
    This function returns the risk level based on the average daily usage hours␣
 ↪passed in the argument.
    It uses if elif else conditions to return the appropriate risk level.
    '''

    if usage_hours < 3:
        return "Low Risk"

    elif 3 <= usage_hours < 6:
        return "Medium Risk"
```

```
    else:
        return "High Risk"

# creating a new column for risk level with the risk level returned from above␣
 ↪function
df['Risk_Level'] = df['Avg_Daily_Usage_Hours'].apply(classify_risk)

print("Risk Level Classification:")
print('-' * 30)
print(df['Risk_Level'].value_counts())
```

```
Risk Level Classification:
------------------------------
Risk_Level
Medium Risk    512
High Risk      150
Low Risk        43
Name: count, dtype: int64
```

### 1.0.10 Preparing Digital detox strategies

[61]:
```
def suggest_detox(risk_level):
    '''
    This function returns digital detox strategies based on the risk level␣
 ↪passed in the argument.
    It is using if elif else conditions to return the appropriate strategy.
    '''

    if risk_level == "High Risk":
        return "Consider significantly reducing usage, setting strict limits,␣
 ↪and seeking professional help if needed."

    elif risk_level == "Medium Risk":
        return "Try setting daily time limits, scheduling screen-free␣
 ↪activities, and being mindful of usage."

    else:
        return "Continue healthy usage habits, be aware of potential triggers,␣
 ↪and maintain a balanced lifestyle."

# creating a new column for Detox suggestions with the values returned from the␣
 ↪above function
df['Detox_Suggestion'] = df['Risk_Level'].apply(suggest_detox)

print("\nDigital Detox Suggestions:")
print('-' * 30)
```

```
# taking a random sample of 5 records to show a the Detox assessment
for index, row in df.sample(5).iterrows():
    print(f"For Student ID {row['Student_ID']} ({row['Risk_Level']}):␣
 ↪{row['Detox_Suggestion']}\n")
```

```
Digital Detox Suggestions:
------------------------------
For Student ID 338 (Medium Risk): Try setting daily time limits, scheduling
screen-free activities, and being mindful of usage.

For Student ID 236 (Medium Risk): Try setting daily time limits, scheduling
screen-free activities, and being mindful of usage.

For Student ID 400 (Medium Risk): Try setting daily time limits, scheduling
screen-free activities, and being mindful of usage.

For Student ID 554 (Medium Risk): Try setting daily time limits, scheduling
screen-free activities, and being mindful of usage.

For Student ID 15 (Medium Risk): Try setting daily time limits, scheduling
screen-free activities, and being mindful of usage.
```

### 1.0.11 Visualizing the data

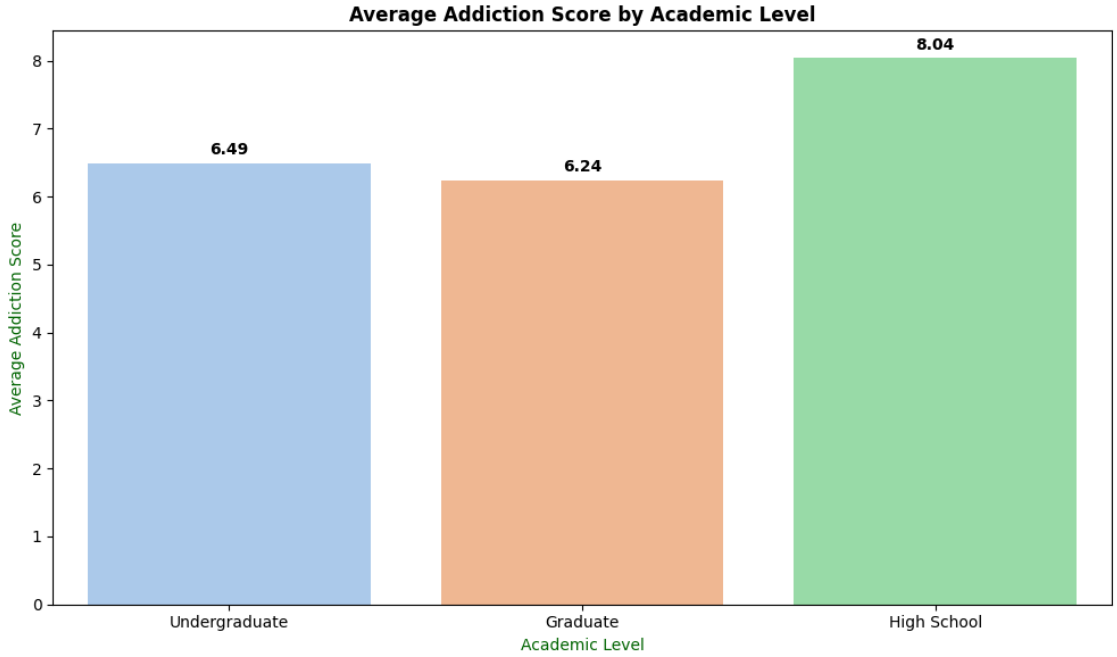
### 1.0.12 Bar chart: Average Addiction Score by Academic Level

[62]:
```
plt.figure(figsize = (10, 6))

# creating a bar plot
ax = sns.barplot(x = 'Academic_Level', y = 'Addicted_Score', data = df, palette␣
 ↪= 'pastel', ci = None)
plt.title('Average Addiction Score by Academic Level', weight = 'bold')
plt.ylabel('Average Addiction Score', color = 'darkgreen')
plt.xlabel('Academic Level', color = 'darkgreen')

# Add labels above the bars with padding
for container in ax.containers:
    ax.bar_label(container, fmt = '%.2f', padding = 3, color = 'black', weight␣
 ↪= 'bold')

plt.tight_layout()
plt.show()
```

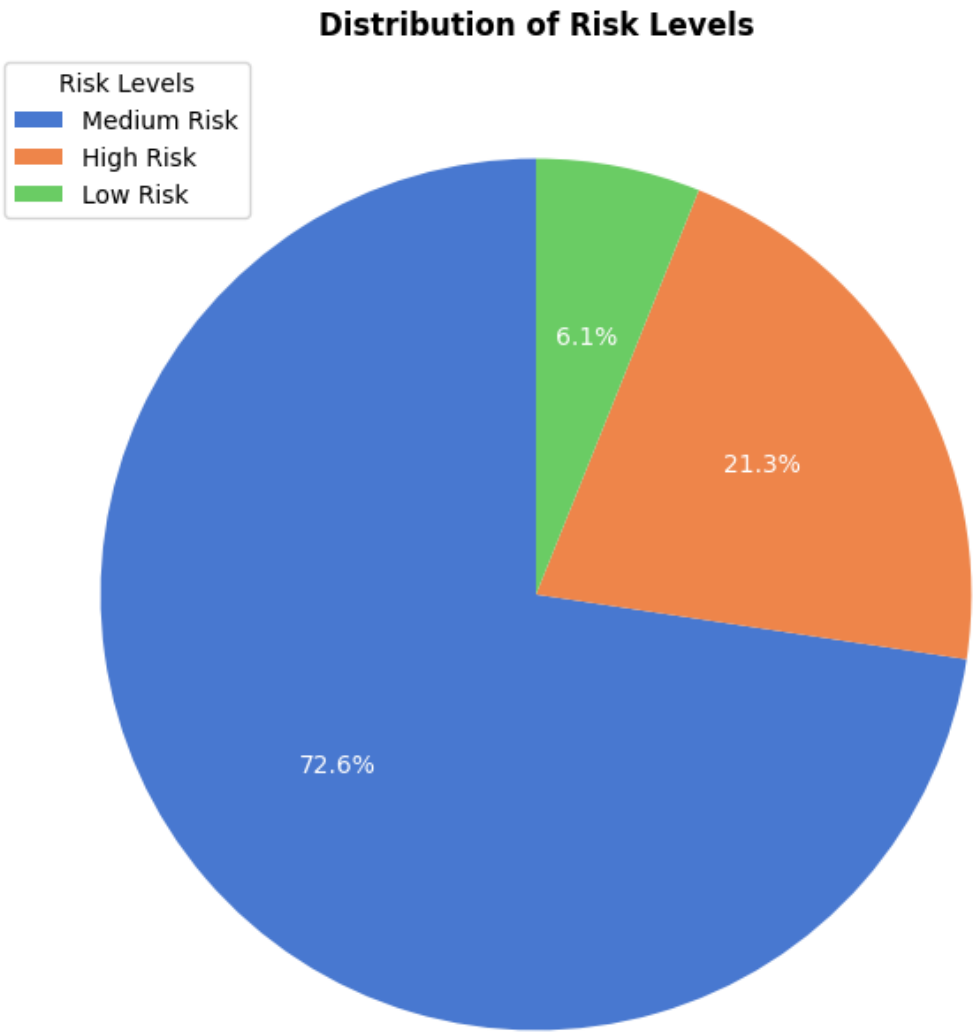



**Bar plot Insight:** - *High School students show the highest average addiction score compared to Undergraduate and Graduate students.* - *Graduate and Undergraduate students have almost similar addiction scores.*

### 1.0.13 Pie chart: Distribution of Risk Levels

[63]:
```
risk_counts = df['Risk_Level'].value_counts() # count of risk level
plt.figure(figsize = (8, 8))

# creating a pie chart
plt.pie(
    risk_counts, labels = risk_counts.index, autopct = '%1.1f%%',
    startangle = 90, colors = sns.color_palette('muted'), textprops={'color':␣
 ↪'white'}
        )
plt.title('Distribution of Risk Levels', weight = 'bold')
plt.legend(title="Risk Levels", loc="upper left", bbox_to_anchor=(0, 1))
plt.show()
```



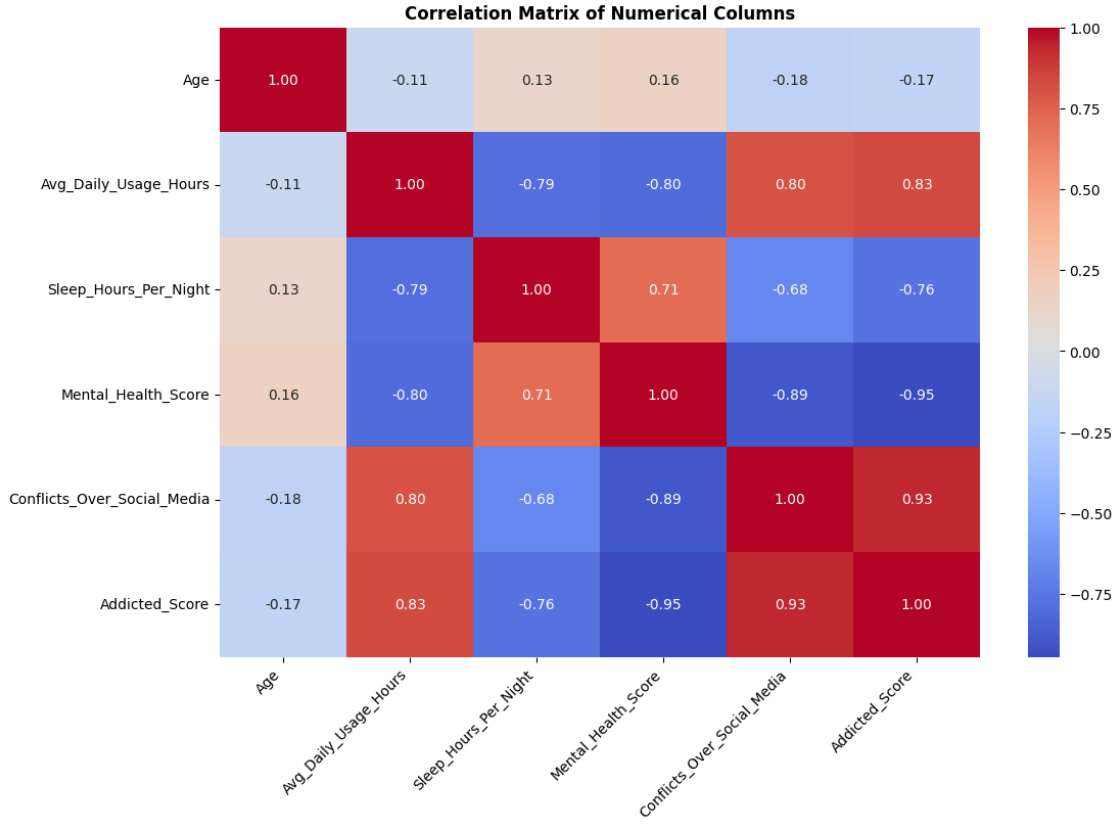

**Pie chart Insight:** - *The "Medium Risk" category represents the largest portion of the student population studied.* - *While "High Risk" is a smaller percentage, it still indicates a significant number of students who may be experiencing more severe issues related to social media usage.* - *The "Low Risk" category is the smallest, suggesting that only a minority of students maintain low social media usage habits.*

### 1.0.14 Heatmap: Correlation matrix of numerical columns

[64]:
```
plt.figure(figsize = (12, 8))

# creating a heat map
sns.heatmap(
```

```
    df[['Age', 'Avg_Daily_Usage_Hours', 'Sleep_Hours_Per_Night',␣
 ↪'Mental_Health_Score', 'Conflicts_Over_Social_Media', 'Addicted_Score']] #␣
 ↪cols to be used
            .corr(), annot = True, cmap = "coolwarm", fmt = ".2f"
            )
plt.title('Correlation Matrix of Numerical Columns', weight = 'bold')
plt.xticks(rotation = 45, ha = 'right') # Rotate x-axis labels
plt.yticks(rotation = 0)
plt.show()
```




**Heat map Insight:** - *When students spend more hours on social media each day, they tend to have higher addiction scores. This is shown by the strong positive correlation (0.93) between Avg_Daily_Usage_Hours and Addicted_Score - These higher addiction scores lead to higher conflicts on social media. This is shown by the strong positive correlation (0.93) between Conflicts_Over_Social_Media and Addicted_Score. - More daily social media usage is linked to getting fewer hours of sleep. This is shown by the strong negative correlation (-0.79) between Sleep_Hours_Per_Night and Avg_Daily_Usage_Hours. - Students with more daily social media usage tend to have lower mental health scores. This is shown by the strong negative correlation (-0.80) between Mental_Health_Score and Avg_Daily_Usage_Hours. - The lowered mental health leads to even more conflicts. This is shown by the strong negative correlation (-0.89) between Mental_Health_Score and Conflicts_Over_Social_Media.*

### 1.0.15 Line plot: Average Daily Usage Hours by Age

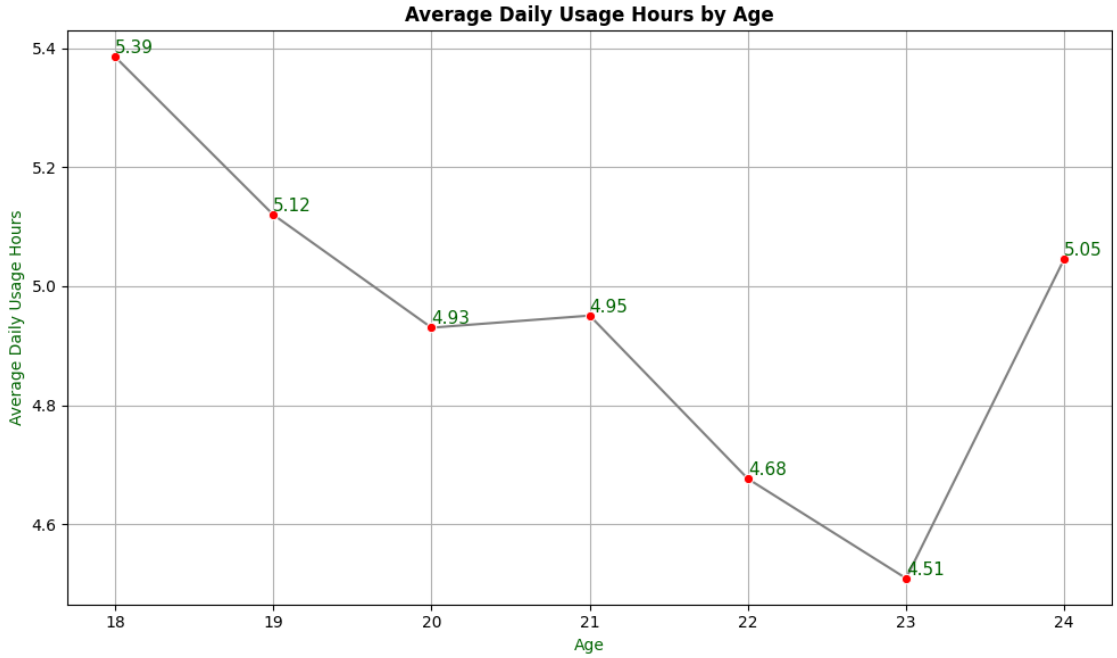
[65]:
```
data = df.groupby('Age')['Avg_Daily_Usage_Hours'].mean().reset_index()
plt.figure(figsize = (10, 6))

# creating a lineplot
ax = sns.lineplot(x = 'Age', y = 'Avg_Daily_Usage_Hours',
             data = data, marker = 'o',
                  markerfacecolor='red', color = 'grey'
             )
plt.title('Average Daily Usage Hours by Age', weight= 'bold')
plt.ylabel('Average Daily Usage Hours', color = 'darkgreen')
plt.xlabel('Age', color = 'darkgreen')
plt.grid(True)

# Add labels to the data points
for x, y in df.groupby('Age')['Avg_Daily_Usage_Hours'].mean().reset_index().
 ↪values:
    plt.text(x, y, f'{y:.2f}', ha = 'left', va = 'bottom', color = 'darkgreen',␣
 ↪fontsize = 11)

plt.tight_layout()

plt.show()
```




**Line plot Insight:** - *Average daily social media usage hours tend to decrease as age people get*

*older, with a significant increase at age 24.*

## 1.1 Summary:

**Here is a summary of the key findings regarding social media addiction among students:**

- *High school students appear to be most susceptible to higher addiction scores, while age shows a general trend of decreasing social media usage with increasing age, although higher usage still correlates with higher addiction scores across all age groups.*
- *Gender does not seem to be a significant factor in addiction levels, with only a slight difference between male (6.3 score) and female (6.5 score) students. Excessive social media use is strongly linked to reduced sleep hours and lower mental health scores.*
- *Furthermore, increased social media engagement is associated with more conflicts in relationships. The study indicates that a majority of students fall into the medium-risk category for social media usage.*
- *The primary issues identified are addiction, interpersonal conflicts, disrupted sleep patterns, and diminished mental well-being, particularly among younger students.*
- *To mitigate these effects, strategies such as setting daily usage limits, engaging in screen-free activities, practicing mindful usage, and seeking professional help for high-risk individuals are recommended.*