INTODUCTION

Education success is influenced not only by intelligence but also by daily habits such as study time, sleep, attendance, screen usage, and lifestyle choices. This project analyses how different student habits affect academic performance using a real-world structured dataset. The analysis is performed using Python, Pandas, NumPy, Matplotlib, and Seaborn with Exploratory Data Analysis (EDA) techniques.

AIM

The main aim of this project is to:

- Understand how daily student habits impact academic performance
- Identify which habits positively or negatively affect grades
- Provide data-driven recommendations to improve student results

Business Problem

Educational institutions often struggle to identify why some students perform better than others despite similar academic resources.

This project helps to:

- Detect performance-influencing habits.
- Support students, teachers, and counselors in designing better academic and wellness strategies.
- Improve overall student success rates through behavioral insights.

Project work flow

1.data collection

“Student Habits vs Academic Performance”, was collected from a structured and reliable source designed for educational research and data analysis. The data represents a combination of academic performance indicators and daily lifestyle habits of students.

Data Source : https://www.kaggle.com/datasets

2.Data understanding

- The dataset contains both numeric and categorical variables.

Key columns include:

- study_hours_per_day,sleep_hours, attendance_percentage
- mental_health_rating, exercise_frequency
- exam_score, screen_time, diet_quality
- Data includes student demographics, lifestyle habits, and academic results.

- Initial checks were done using .shape(), .info(), and .describe().

3.Data cleaning

The following steps were performed:

- Missing values handled using mean/median/mode where appropriate.
- Duplicate records were identified and removed.
- Outliers in numerical columns such as exam_score and screen_time were detected using boxplots.
- Categorical inconsistencies were cleaned (e.g., inconsistent text labels).
- To find the missing values

  df.isnull().sum()

- To check the duplicate values and sum

  df.duplicated().sum()

  #MISSING VALUE HANDLING

```
# Numeric columns
   numeric_cols = [
      'age', 'study_hours_per_day', 'social_media_hours',
```

```
    'netflix_hours', 'attendance_percentage', 'sleep_hours',
    'exercise_frequency', 'mental_health_rating', 'exam_score'
]

for col in numeric_cols:
    df[col] = df[col].fillna(df[col].mean())



# Categorical columns categorical_cols = [
    'gender', 'part_time_job', 'diet_quality',
    'parental_education_level', 'internet_quality',
    'extracurricular_participation'
]

for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])
```

#fix invalid data ranges

```
df = df[(df['attendance_percentage'] >= 0) &
    (df['attendance_percentage'] <= 100)]
df = df[(df['sleep_hours'] >= 0) & (df['sleep_hours'] <= 12)]
```

#standardize coloumn name

```
df.columns = df.columns.str.lower().str.replace(" ", "_")
```

```
#fixing datatypes
df['age'] = df['age'].astype(int)
df['exam_score'] = df['exam_score'].astype(float)
df['exercise_frequency'] = df['exercise_frequency'].astype(int)
df['mental_health_rating'] = df['mental_health_rating'].astype(int)

# remove_outliers_iqr(df, col):
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    return df[(df[col] >= lower) & (df[col] <= upper)]

df = remove_outliers_iqr(df, 'study_hours_per_day')
df = remove_outliers_iqr(df, 'social_media_hours')
df = remove_outliers_iqr(df, 'exam_score')lier detection
def remove_outliers_iqr(df, col):
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
```

```
    return df[(df[col] >= lower) & (df[col] <= upper)]

df = remove_outliers_iqr(df, 'study_hours_per_day')
df = remove_outliers_iqr(df, 'social_media_hours')
df = remove_outliers_iqr(df, 'exam_score')

# should be all zeros
df.isnull().sum()
df.duplicated().sum()
df.info()
df.describe()
```

7.Derived Metrics

New columns created:

New Column	Purpose
Study_Efficiency	Exam_Score / Study_Hours
Performance_Level	Low / Medium / High
Digital_Addiction	Based on Screen_Time
Sleep_Category	Poor / Adequate / Excess

```
df['total_screen_time'] = df['social_media_hours'] +
df['netflix_hours']
```

```
df['study_sleep_ratio'] = df['study_hours_per_day'] /
df['sleep_hours']
df.head()
```

8. Filtering data

```
df['academic_efficiency'] = df['exam_score'] /
df['study_hours_per_day']


def performance_label(score):
    if score < 50:
        return "Low"
    elif score < 75:
        return "Medium"
    else:
        return "High"

df['performance_level'] = df['exam_score'].apply(performance_label)

# separate numerical and categorical data
cat_cols=df.select_dtypes(include=['object']).columns
num_cols = df.select_dtypes(include=np.number).columns.tolist()
print("Categorical Variables:")
print(cat_cols)
print("Numerical Variables:")
```

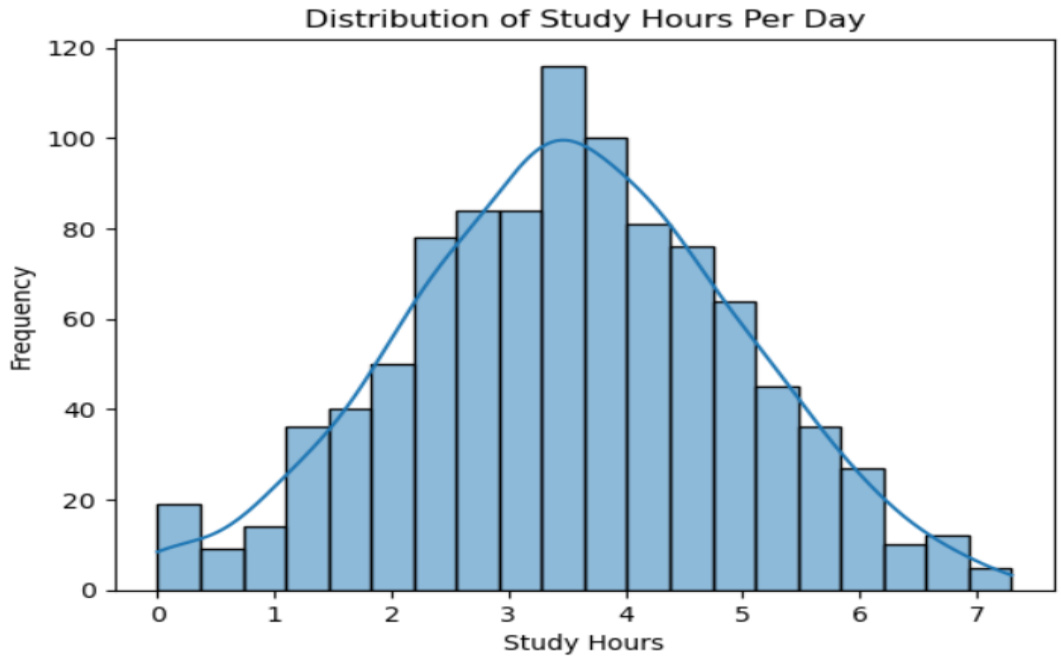
```
print(num_cols).


#dropping coluomns
df.drop("student_id",axis=1,inplace=True)
df.head()
df.drop(["social_media_hours","netflix_hours"],axis=1,inplace=True)
```

9.EDA

UNIVARIATE ANALYSIS

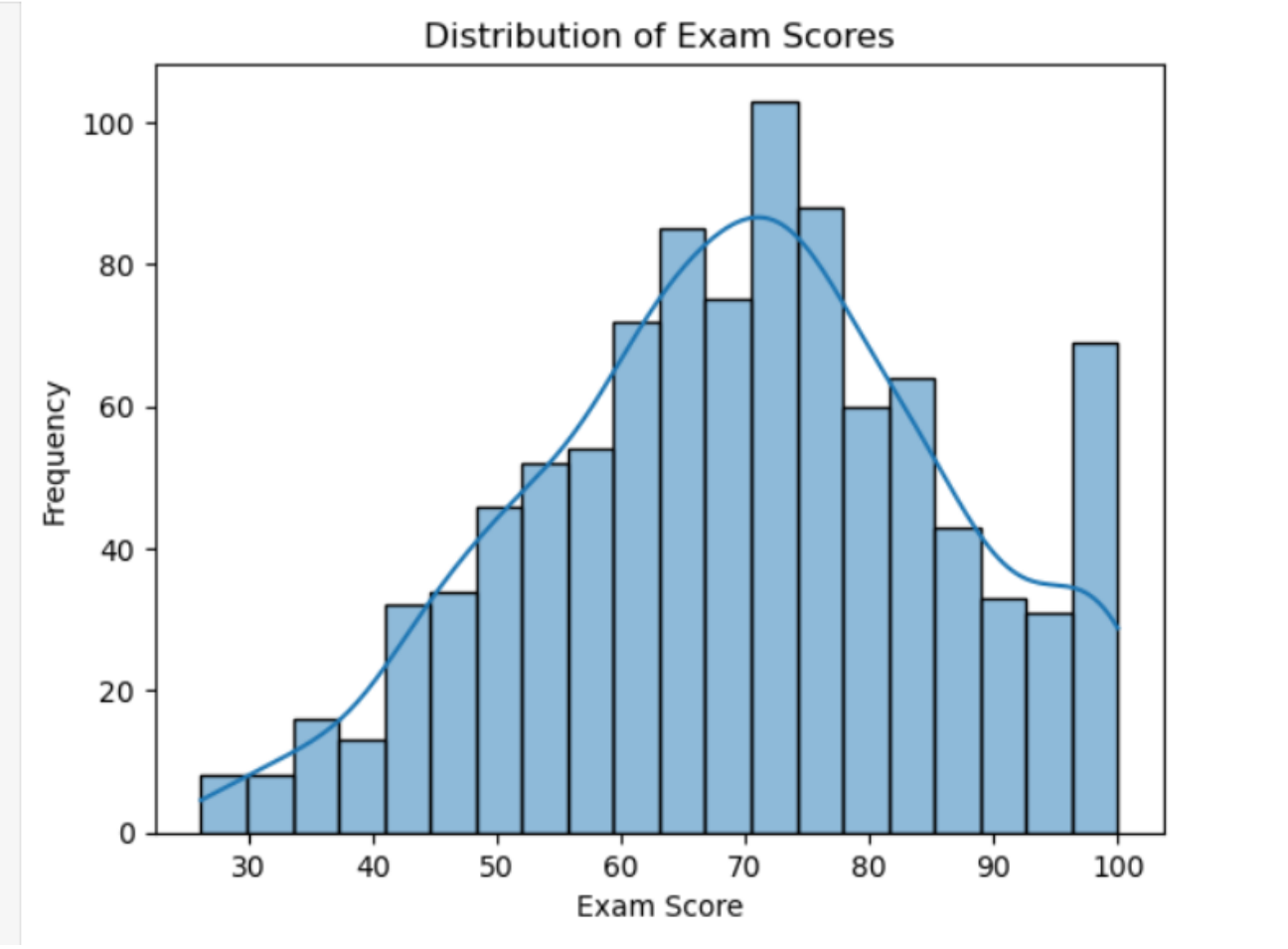
- Distribution of Exam Scores
- Distribution of Study Hours
- Distribution of Sleep Hours

```
plt.figure()
sns.histplot(df['study_hours_per_day'], bins=20, kde=True)
plt.title("Distribution of Study Hours Per Day")
plt.xlabel("Study Hours")
plt.ylabel("Frequency")
plt.show()
```



#most of the students study between 2-5 hours per day

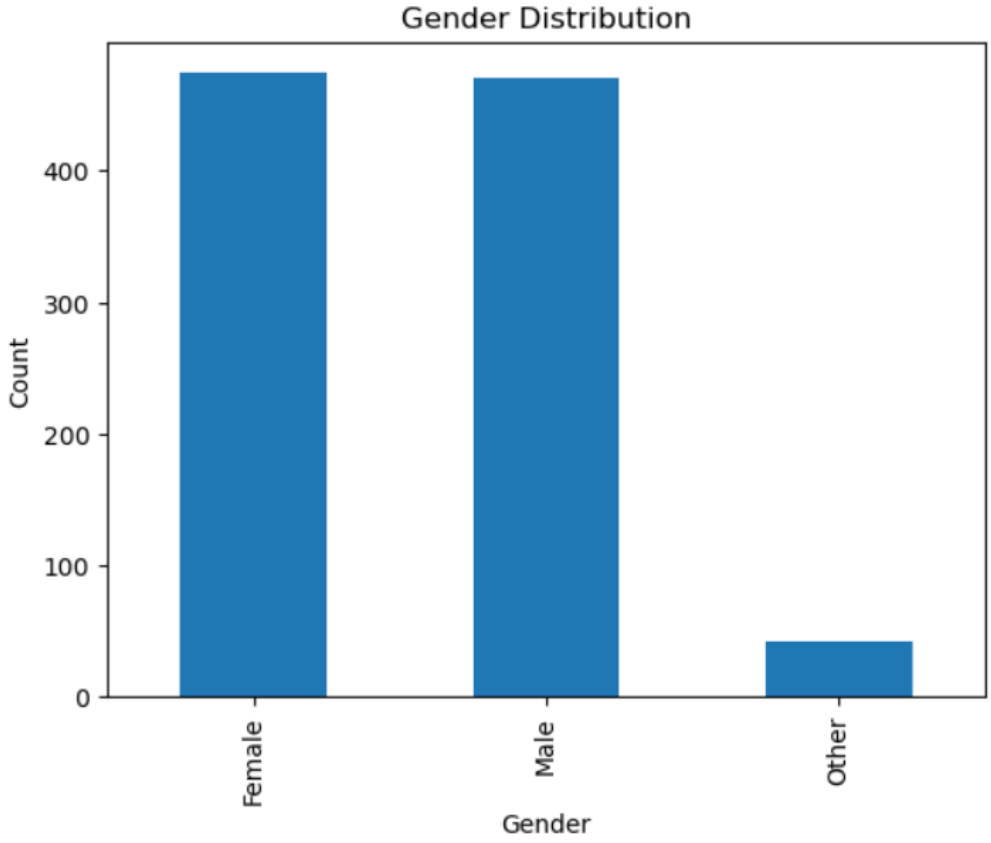
```
plt.figure()
sns.histplot(df['exam_score'], bins=20, kde=True)
plt.title("Distribution of Exam Scores")
plt.xlabel("Exam Score")
plt.ylabel("Frequency")
plt.show()
```



#majority of the students scored between 50-80 marks

```
plt.figure()
df['gender'].value_counts().plot(kind='bar')
plt.title("Gender Distribution")
plt.xlabel("Gender")
plt.ylabel("Count")
```

plt.show()



#the gender is nearly balanced

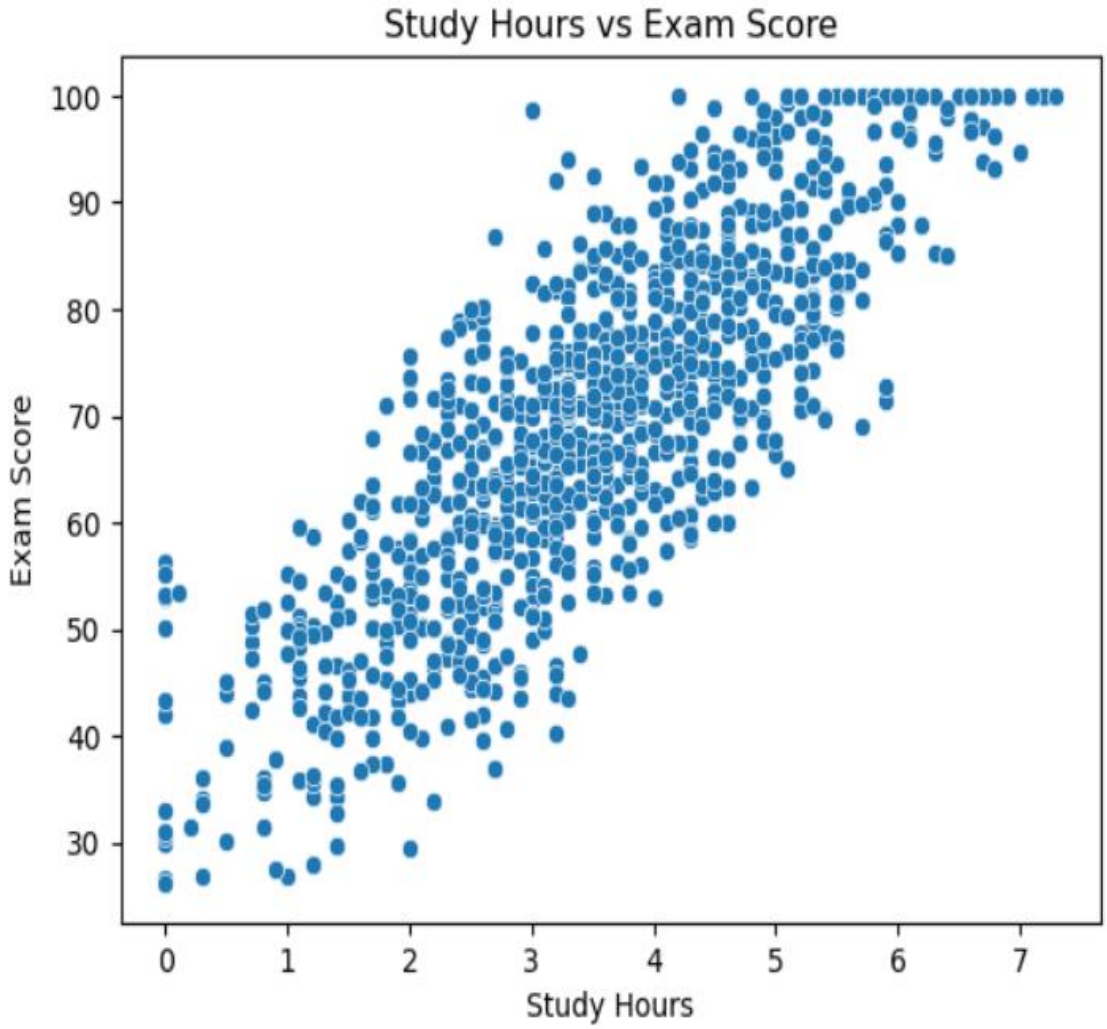
BIVARIATE ANALYSIS

- Study Hours vs Exam Score
- Sleep Hours vs Exam Score
- Screen Time vs Exam Score

plt.figure()

sns.scatterplot(x='study_hours_per_day', y='exam_score', data=df)

```
plt.title("Study Hours vs Exam Score")
plt.xlabel("Study Hours")
plt.ylabel("Exam Score")
plt.show()
```



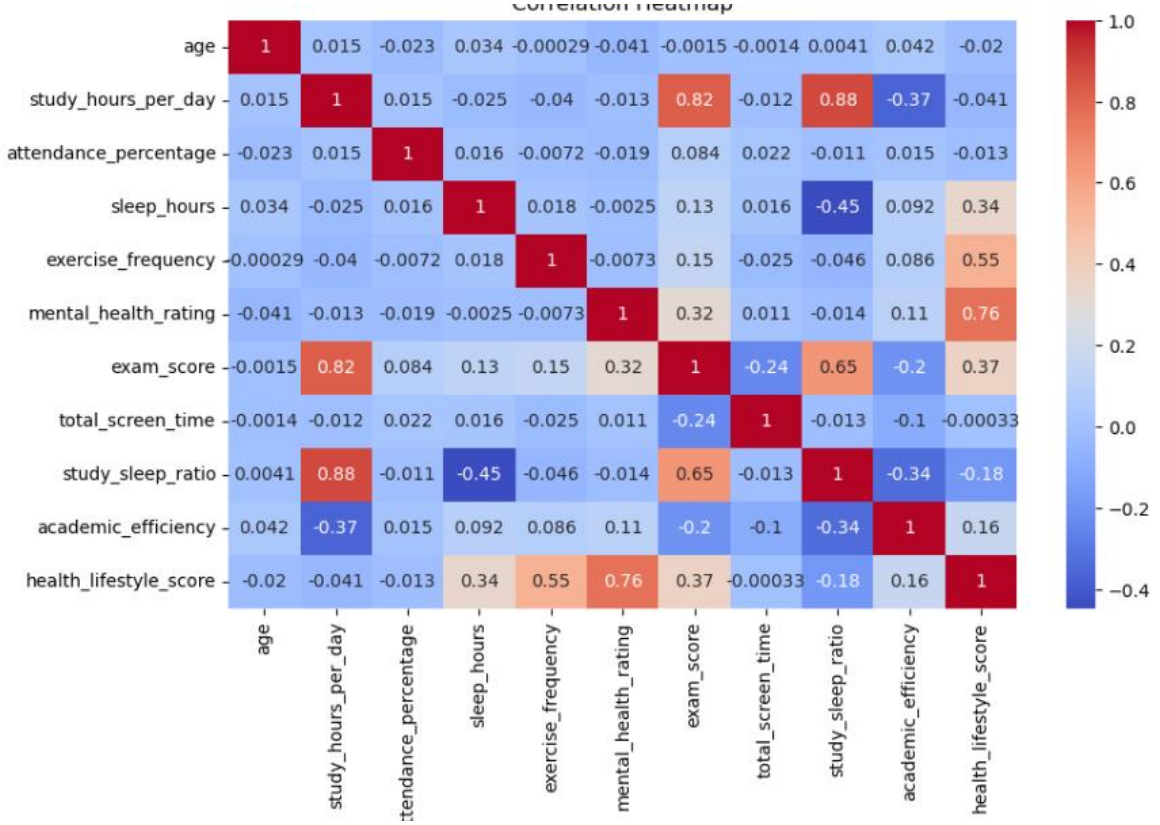
#positive correlation

#more study hours generally leads to higher marks

MULTIVARIATE ANALYSIS

- Study Hours, Sleep Hours & Exam Score
- Attendance & Exam Performance by Parental Support

```
plt.figure(figsize=(10,6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



11. Key Insights

- Students studying 3–5 hours daily perform best
- Sleep below 6 hours leads to poor academic performance
- High attendance (>85%) strongly improves exam scores
- Excessive screen time (>6 hours) reduces performance
- Parental support significantly boosts performance
- Students with low stress & balanced habits achieve higher grades

CONCLUSION

- This project clearly shows that academic performance is highly influenced by student habits. Proper study time, quality sleep, attendance, and controlled screen use result in better academic outcomes.
- Students who study more hours per day score significantly higher in exams.
- More screen time leads to lower academic performance.