Name: Charity V. Katokwe

Term Project

Student Habits Performances Report

1. Introduction

i.) Problem Statement

Academic performance is influenced by multiple behavioral and lifestyle factors. However, quantifying this influence and making predictions based on it remains a challenge.

ii.) Project goals

- Explore and understand the impact of student habits (e.g., study hours, social media usage) on exam performance.
- Build predictive models to estimate exam scores based on these habits.
- Compare model performance and derive actionable insights.

iii.) Importance of Topic

Understanding what drives academic success can help students optimize their routines and assist educators in shaping support strategies.

2. Dataset selection

For this project, I chose a dataset focused on student-habits-performance. The selected dataset contains data on 1,000 students and includes a wide variety of lifestyle, academic, and demographic variables. These include study time, social media use, exercise frequency, sleep hours, attendance, and parental education, as well as the students' final exam scores. This data allows for a holistic analysis of how different daily habits and environmental factors affect academic performance.

This dataset is highly relevant in real-world educational contexts where understanding the drivers of student success is critical. It provides an opportunity to explore questions such as: “Do students who sleep more perform better?”, “Is social media usage negatively

correlated with grades?", and "Does having a part-time job impact exam score?" The dataset is non-trivial in size and structure, making it well-suited for statistical analysis, machine learning modeling, or predictive analytics.

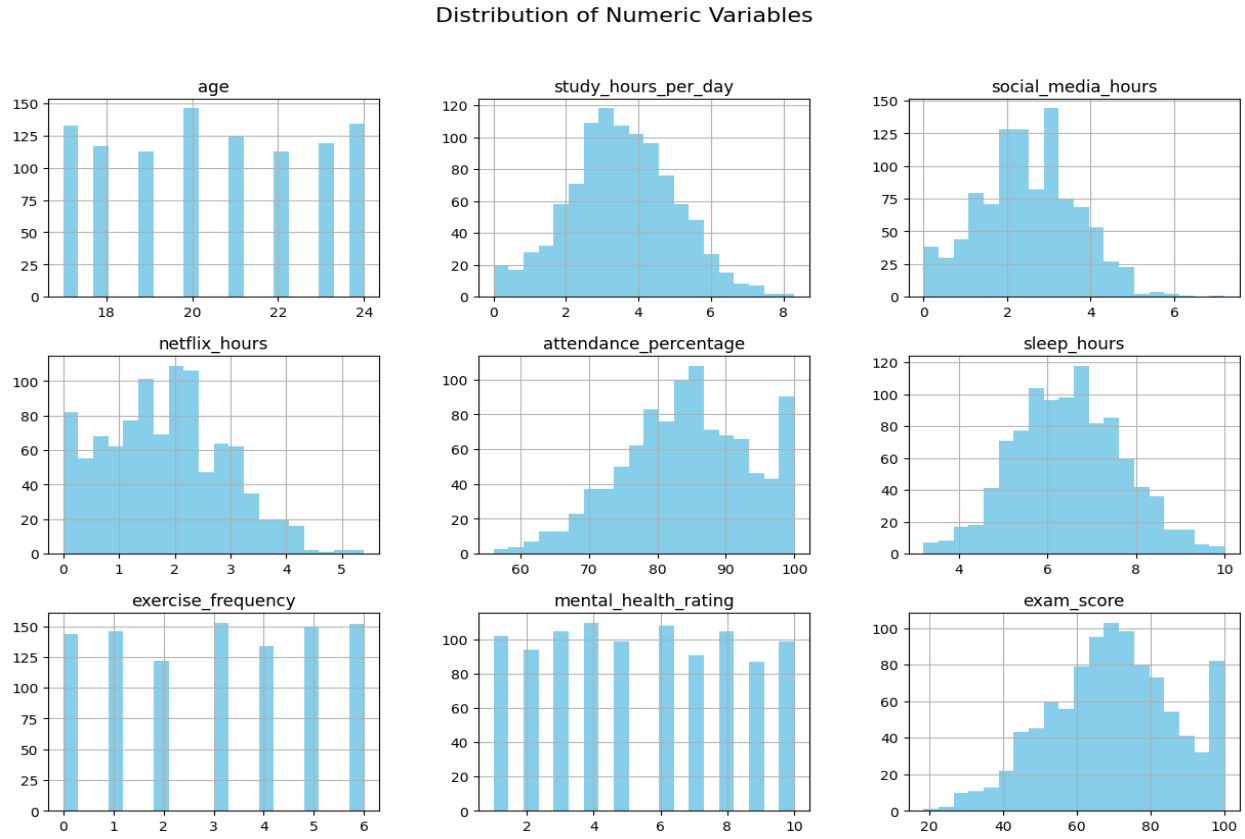
3. Methodology

The dataset has no missing value, so it was ready for analysis. It consists of 16 columns including numeric (study-hours-per-day, attendance-percentage, exam-score and many others) and categorical (gender, diet-quality, internet-quality and others) information. I used **Google Colab** platform. For python libraries that I have used, they include:

- Python.
- Pandas, numpy – data manipulation.
- Matplotlib, seaborn – data visualization.
- Sklearn – machine learning models and evaluation.

4. Exploratory Data Analysis (EDA)

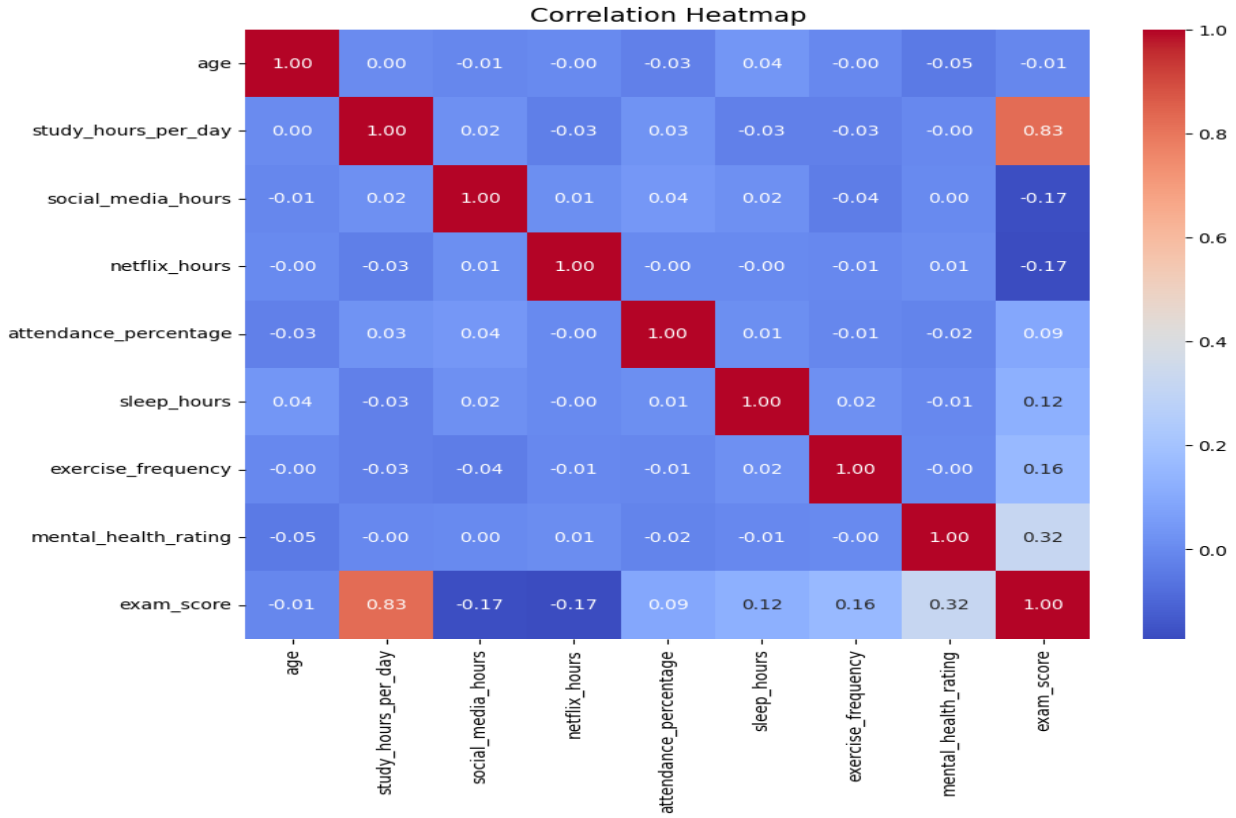
i.) Histograms



Key observations:

- **Age** – This distribution appears uniform across the observed ages, suggesting a relatively even representation of individuals within this age range in the dataset.
- **Study hours per day** – This distribution is skewed to the right, indicating that most individuals tend to study for a smaller number of hours per day, with fewer people studying for longer durations. The peak is around 3-4 hours
- **Social media hours** – This distribution also shows a positive skew (skewed to the right). Like study hours, most individuals spend less time on social media, but a few spend considerably more. The mean is likely greater than the median.
- **Netflix hours** – Again, we see a positive skew (skewed to the right). Most individuals watch Netflix for a shorter duration, with a smaller group engaging for longer periods. The mean is likely greater than the median.
- **Attendance percentage** – This distribution displays a negative skew (skewed to the left). The tail of the distribution extends towards the lower values, indicating that most individuals have high attendance, with fewer individuals having significantly lower attendance. The mean is likely less than the median.
- **Sleep hours** – This distribution looks somewhat like a normal distribution, centered around 6-8 hours of sleep. This suggests that most individuals in the dataset get a moderate amount of sleep, with fewer individuals getting significantly more or less sleep.
- **Exercise frequency** - This distribution appears uniform across the observed frequencies, like the age distribution. This suggests a relatively even representation of different exercise frequencies in the dataset.
- **Mental health rating -** This distribution also seems relatively uniform across the different mental health ratings, indicating an even spread of self-reported mental health within the observed range.
- **Exam Score** – This distribution appears somewhat like a normal distribution, centered around the 70-80 range. This suggests that most individuals achieved scores in this range, with fewer individuals scoring significantly higher or lower. There's a slight peak around the 90-100 range as well.

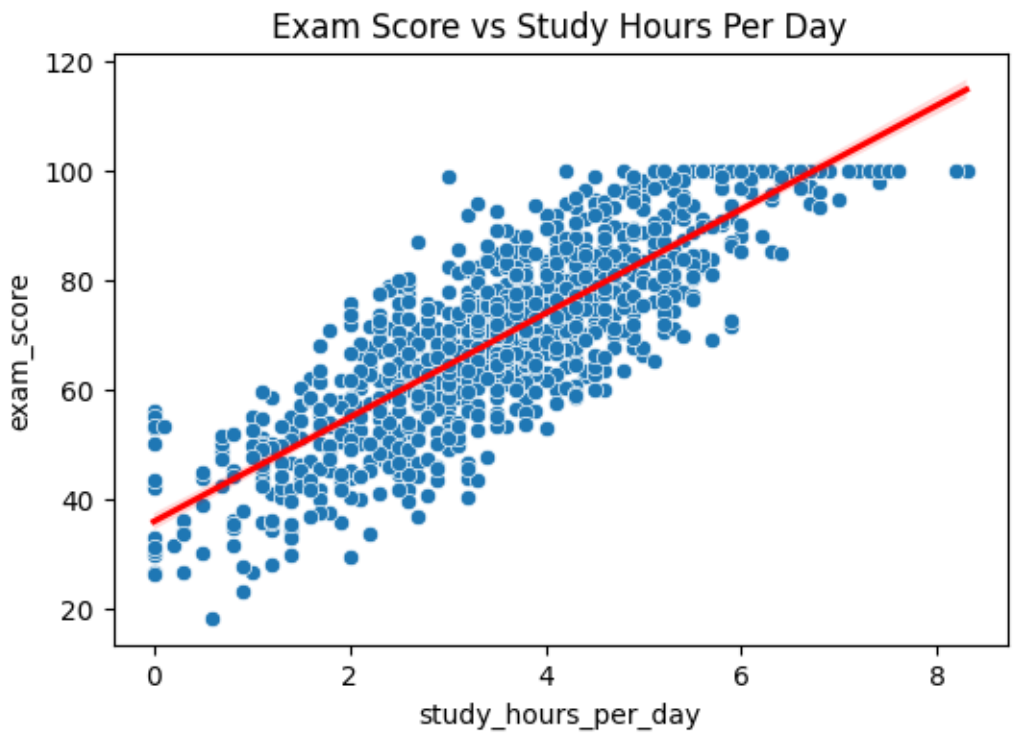
ii.) Correlation heatmap



Key observations:

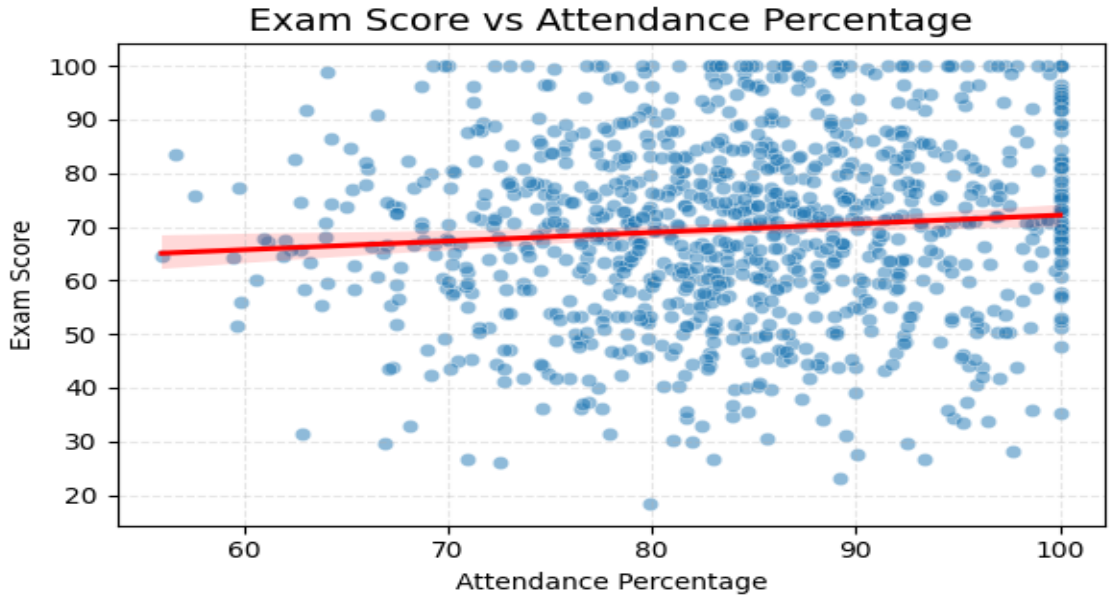
- There is a strong positive correlation (0.83) between study hours per day and exam scores. This suggests that students who study more tend to achieve higher exam scores.
- There appears to be a moderate positive correlation (0.32) between mental health rating and exam score, suggesting that students with better self-reported mental health tend to perform slightly better on exams.
- There's a weak negative correlation (-0.17) between social media hours and exam score, and a weak negative correlation (-0.17) between Netflix hours and exam score. This hints that perhaps more time spent on social media or watching Netflix might be associated with slightly lower exam scores, although the relationship isn't very strong.
- Most other correlations appear to be quite weak (close to zero), suggesting little to no linear relationship between those pairs of variables. For example, age seems to have very little correlation with most other factors.

iii.) Scatter plots with regression line



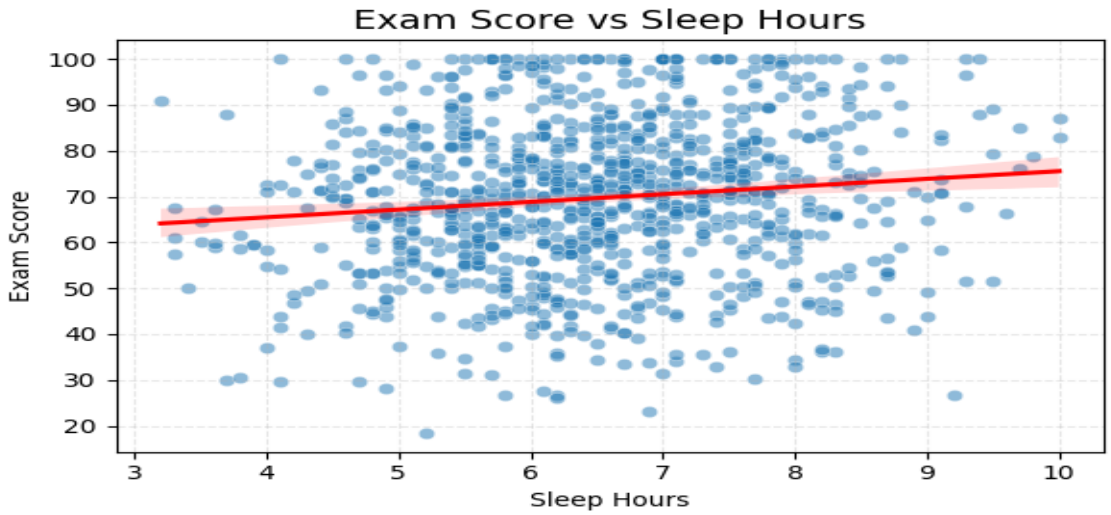
Key observations:

- This scatter plot strongly suggests a positive correlation between the number of hours spent studying per day and the resulting exam scores. The regression line provides a visual representation of this trend, indicating that more study time is associated with higher exam performance, on average.



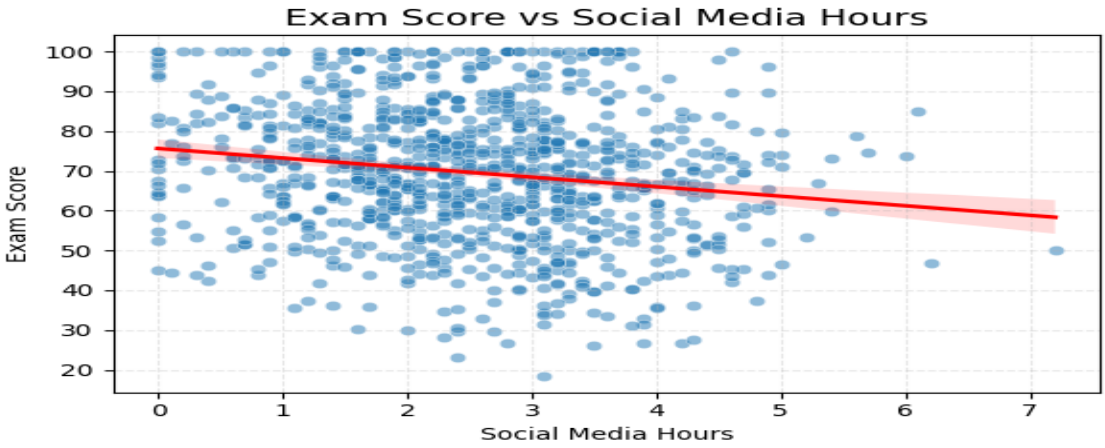
Key observations:

- This scatter plot suggests that, in this dataset, the attendance percentage alone is not a strong predictor of exam scores. While there might be a very slight tendency for higher attendance to be associated with slightly higher scores on average, the relationship is weak, and many other factors are likely at play.



Key observations:

- This scatter plot suggests a slight tendency for more sleep to be associated with somewhat higher exam scores on average, the relationship is weak. There's a lot of variability in the data, indicating that other factors likely have a much stronger influence on exam performance than just the number of hours slept.



Key observations:

- This scatter plot suggests a slight tendency for more time spent on social media to be associated with somewhat lower exam scores on average, but the relationship is weak. The considerable scatter in the data indicates that other factors likely play a more significant role in determining exam performance.

5. Data cleaning and processing

	0
student_id	0
age	0
gender	0
study_hours_per_day	0
social_media_hours	0
netflix_hours	0
part_time_job	0
attendance_percentage	0
sleep_hours	0
diet_quality	0
exercise_frequency	0
parental_education_level	91
internet_quality	0
mental_health_rating	0
extracurricular_participation	0
exam_score	0

dtype: int64

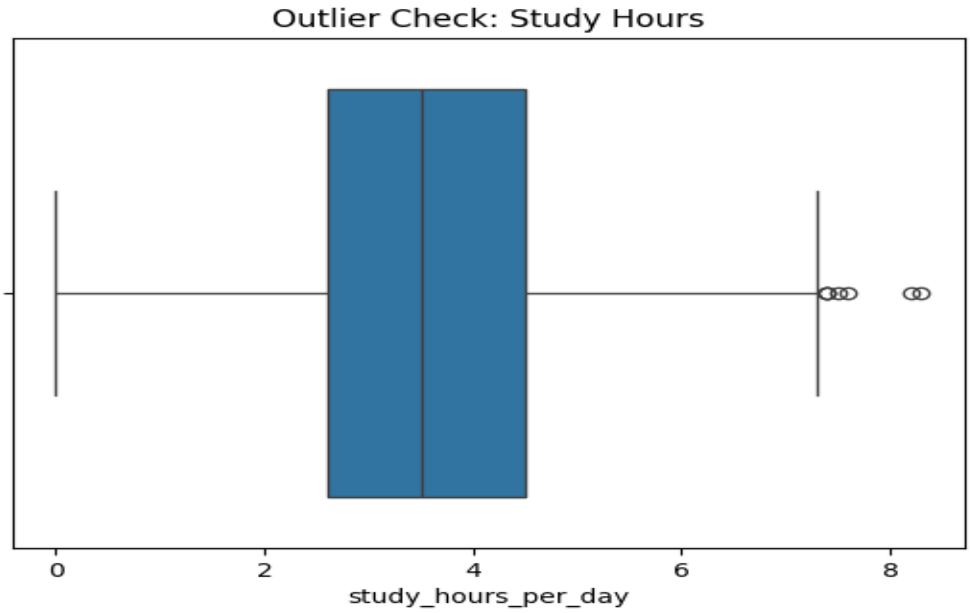
```
#Imputation strategy
df['parental_education_level'] = df['parental_education_level'].fillna(df['parental_education_level'].mode()[0])

```

My data cleaning process has revealed that I have a clean dataset with no missing values in most columns. However, the parental-education-level column has 91 missing entries that I will need to address using an appropriate data imputation or handling technique before proceeding with my analysis or modeling.

I filled missing values in the parental-education-level column to ensure the dataset is complete and suitable for modeling. Since it's a categorical feature, I used the most common category or a placeholder like "Unknown" to avoid dropping data. This helps maintain consistency and allows machine learning algorithms to process data without errors.

i.) Outliers



This box plot confirms the distribution of study hours and clearly identifies several data points that could be considered outliers due to their unusually high or low values compared to the rest of the data

study_hours_per_day

```
Q1 = df['study_hours_per_day'].quantile(0.25)
Q3 = df['study_hours_per_day'].quantile(0.75)
IQR = Q3 - Q1

upper_limit = Q3 + 1.5 * IQR

# Cap values above upper limit
df['study_hours_per_day'] = df['study_hours_per_day'].apply(
    lambda x: upper_limit if x > upper_limit else x
)

```

I used the Interquartile Range (IQR) method to detect and cap outliers in the study-hours-per-day column. It calculates the upper limit as Q3 + 1.5 * IQR and replaces any value above this limit with the upper limit itself. This approach reduces the influence of extremely high values while preserving overall data distribution.

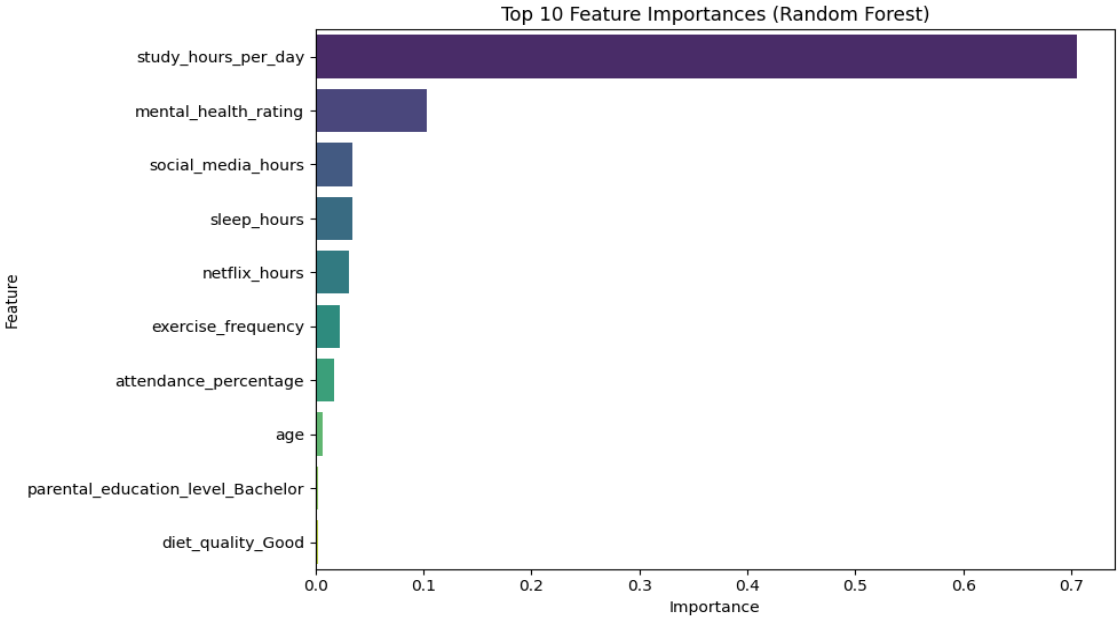
6. Identifying Business Analytics Questions

- What are the key behavioral factors that influence a student's academic performance?
- Can we predict exam scores based on lifestyle and academic habits?
- How does attendance affect student performance, and is it more important than study?
- What combination of factors can be used to identify at-risk students early?

7. Building predictive models and evaluation

i.) Random forest model

I used Random Forest model because it handles both numerical and categorical data well and captures non-linear relationships. It also provides reliable feature importance insights for identifying key performance drivers.



The Random Forest model identified the most important predictors of exam performance based on feature importance scores. Key features like study-hours-per-day, mental-health-

rating and social-media-hours had the highest influence. These insights help focus academic interventions on the most impactful student habits

ii.) Linear regression modelling

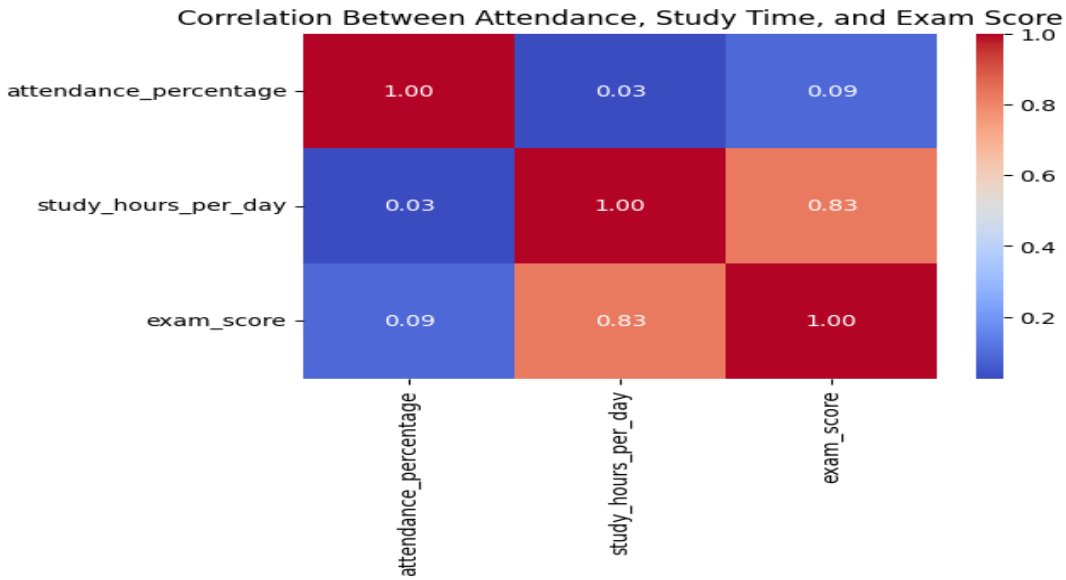
I used the Linear Regression model because it is simple, interpretable, and effective for predicting a continuous outcome like exam scores based on various student habits and attributes.

```
Linear Regression - RMSE: 5.14
Linear Regression - R² Score: 0.90
```

Evaluation showed an RMSE of approximately [**5.14]** and an R^2 score of **[0.90]**, indicating the model's accuracy and variance explained

iii.) Correlation Analysis

I used **correlational analysis** because it helps quantify the strength and direction of the relationship between variables specifically, how closely attendance and study time are related to exam scores. This method is ideal for identifying which factors have a stronger linear influence before applying more complex models.



The heatmap shows a strong positive correlation (0.83) between study-hours-per-day and exam-score. There is a weak positive correlation (0.09) between attendance-percentage and exam-score. The correlation between attendance-percentage and study-hours-per-day is also very weak (0.03).

8. Key findings

- Study-hours are the strongest predictor of exam performance followed by mental-health-rating.
- Time spent on social media and Sleeping had a medium impact on students' performance.
- Surprisingly, Parental education level had a mild impact, suggesting some background influence, though not as strong as daily habits.
- Students with better diet habits tended to perform better, though the effect was less strong.

9. Summary

Behavioral factors like study hours and attendance play a crucial role in student performance. Random Forest models provided the most accurate predictions with an R^2 of ~0.90

10. Recommendations

- **Encourage consistent study habits**

  Institutions and educators should promote structured daily study routines, as study hours were the strongest predictor of exam performance.
- **Support mental health initiatives**

  Since mental health significantly impacts performance, schools should invest in counseling services, stress management workshops, and mindfulness programs.
- **Promote digital well-being**

  Moderate social media and screen usage should be encouraged. Digital literacy sessions can help students manage time spent on entertainment platforms.

- **Educate students on nutrition**
  Although diet has a smaller effect, awareness campaigns around healthy eating may further support academic performance and overall well-being.
- **Focus on student-driven success**
  With parental education showing only mild influence, resources should be directed toward empowering students through self-discipline, peer mentoring, and study skills training.

11. Limitations

- The dataset may not be representative of all students (sample bias).
- Time-series or longitudinal patterns were not explored.

12. References

https://www.kaggle.com/datasets/jayaantanaath/student-habits-vs-academic-performance

Appendix

Importing the dataset using pandas library of the python programming language.

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

```
Choose Files  student_ha...ormance.csv
• student_habits_performance.csv(text/csv) - 73663 bytes, last modified: 5/3/2025 - 100% done
Saving student_habits_performance.csv to student_habits_performance.csv
```

```
[2]  1 import pandas as pd
     2
     3 # For CSV
     4 df = pd.read_csv("student_habits_performance.csv")
     5
     6 # For Excel
     7 # df = pd.read_excel("your_file_name.xlsx")
     8
```

Exploring the type of data we have for each column

```
[3]  1 import matplotlib.pyplot as plt
```

```
1 print(df.info())
2 print(df.isnull().sum())
3
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   student_id                    1000 non-null   object
 1   age                           1000 non-null   int64
 2   gender                        1000 non-null   object
 3   study_hours_per_day           1000 non-null   float64
 4   social_media_hours            1000 non-null   float64
 5   netflix_hours                 1000 non-null   float64
 6   part_time_job                 1000 non-null   object
 7   attendance_percentage         1000 non-null   float64
 8   sleep_hours                   1000 non-null   float64
 9   diet_quality                  1000 non-null   object
 10  exercise_frequency            1000 non-null   int64
 11  parental_education_level      909 non-null    object
 12  internet_quality              1000 non-null   object
 13  mental_health_rating          1000 non-null   int64
 14  extracurricular_participation 1000 non-null   object
 15  exam_score                    1000 non-null   float64
dtypes: float64(6), int64(3), object(7)
memory usage: 125.1+ KB
None
student_id                    0
age                           0
gender                        0
```

✓ 0s completed at 11:22 AM

Predictive modelling methodologies (linear regression and random forest comparison table)

```
[19] 1 from sklearn.linear_model import LinearRegression
     2 from sklearn.ensemble import RandomForestRegressor
     3 from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
     4 import numpy as np
     5 import pandas as pd
     6
```

Define a Helper Function to Evaluate Models

```
[22] 1 def evaluate_model(name, model, X_train, X_test, y_train, y_test, results):
     2     model.fit(X_train, y_train)
     3     y_pred = model.predict(X_test)
     4
     5     rmse = np.sqrt(mean_squared_error(y_test, y_pred))
     6     mae = mean_absolute_error(y_test, y_pred)
     7     r2 = r2_score(y_test, y_pred)
     8
     9     results.append({
    10         'Model': name,
    11         'RMSE': round(rmse, 2),
    12         'MAE': round(mae, 2),
    13         'R² Score': round(r2, 3)
    14     })
    15
```

Train and compare models

```python
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
import pandas as pd
import numpy as np

results = []

# Drop non-numeric columns from training/testing data
X_train = X_train.select_dtypes(include=[np.number])
X_test = X_test.select_dtypes(include=[np.number])

# Linear Regression
lr = LinearRegression()
evaluate_model("Linear Regression", lr, X_train, X_test, y_train, y_test, results)

# Random Forest
rf = RandomForestRegressor(random_state=42)
evaluate_model("Random Forest", rf, X_train, X_test, y_train, y_test, results)

# Convert results to DataFrame and sort by RMSE
results_df = pd.DataFrame(results)
results_df = results_df.sort_values(by="RMSE")

# Display the sorted results
print(results_df)


```

```
               Model  RMSE   MAE  R² Score
0  Linear Regression  5.08  4.12     0.899
1      Random Forest  6.12  4.90     0.854
```