INT 234

PREDICTIVE ANALYTICS

Project Name

Student Exam Score Predictor

SUBMITTED BY

Name– Kushagra

Registration Number – 12323868

Roll No.- 06

Under the Guidance of-

Dr. Tanima Thakur Ma’am

School of Computer Science & Engineering

DECLARATION

I Kushagra, student of under CSE/IT Discipline at Lovely Professional University, Punjab, hereby declare that the work presented in this project report is my original work and has not been submitted elsewhere for any degree or diploma.

Date: 16-Dec-2025
Signature: Kushagra
Registration No: 12323868

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide Dr. Tanima Thakur for their valuable guidance and continuous support throughout this project.

I am also thankful to Lovely Professional University for providing the necessary resources and learning environment. Lastly, I thank my friends and family for their encouragement.

TABLE OF CONTENTS

1. INTRODUCTION

Education systems increasingly rely on data-driven approaches to evaluate and improve student performance. Predicting exam scores in advance helps teachers, students, and institutions take corrective actions such as additional practice, counseling, or academic support.

The Student Exam Score Predictor project is designed to analyze various academic parameters of students and predict their final exam scores using Python and Machine Learning techniques. This project focuses on understanding how factors like study hours, attendance, and previous academic performance collectively influence examination results.

The project also emphasizes Exploratory Data Analysis (EDA) and visualization techniques to extract meaningful insights from the dataset. A regression-based machine learning model is implemented to make accurate predictions and validate assumptions.

Objectives of the Project:

- To analyze student performance data
- To identify key factors affecting exam scores
- To build a predictive model using Machine Learning
- To visualize patterns and relationships in the dataset

2. SOURCE OF DATASET

In this project, the dataset is manually generated to simulate real-world student performance data. The data is created by considering common academic factors that influence exam results. Since real institutional data is often confidential and not publicly accessible, a synthetic dataset is designed to closely resemble actual student records.

Dataset Attributes Include:

- Study Hours per Day
- Attendance Percentage
- Diet Quality
- Previous Exam Scores
- Assignment / Internal Assessment Marks
- Exam Score
- Final Exam Score (Target Variable)

The dataset is stored in CSV format and imported into Python using libraries such as Pandas and NumPy.

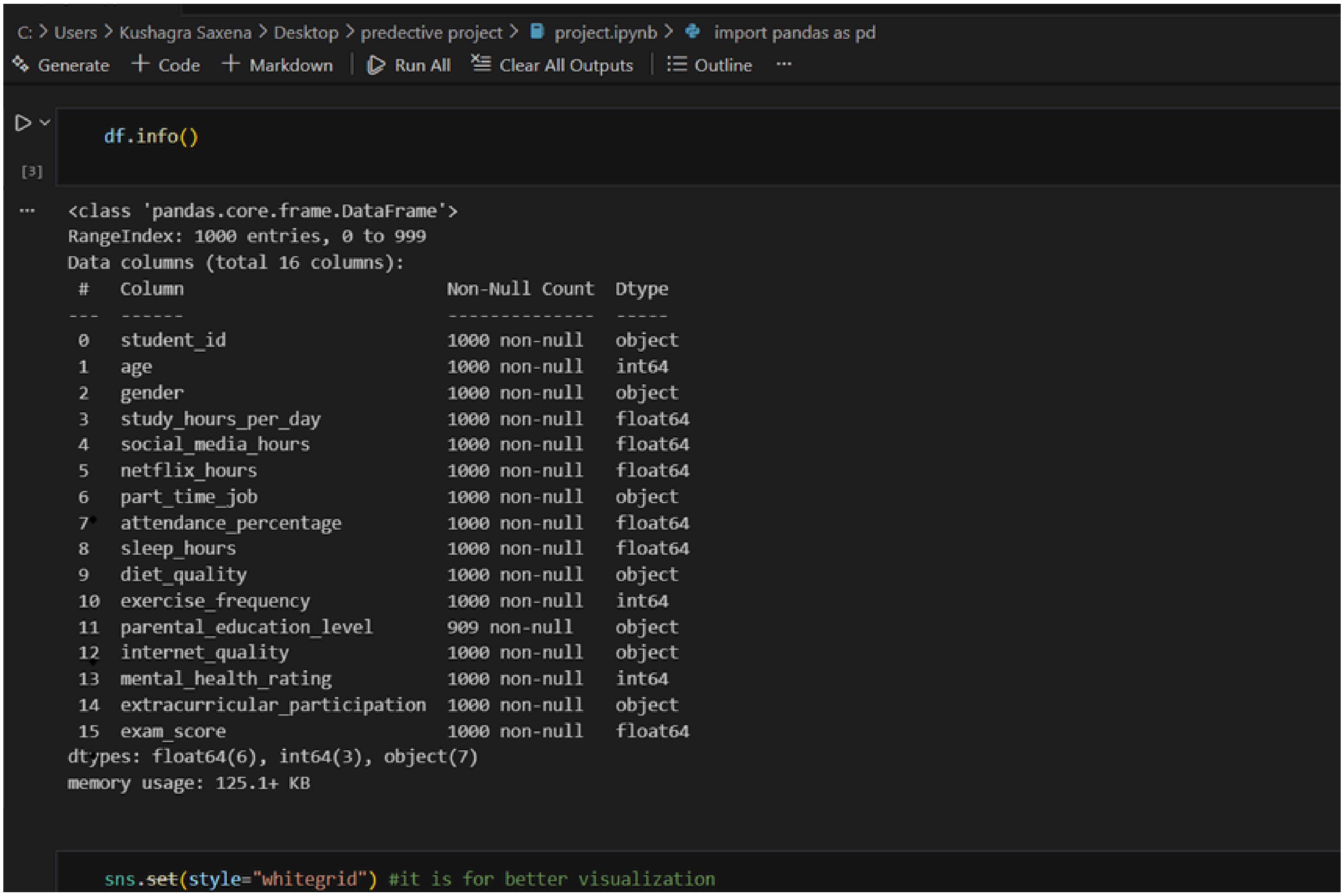
3. DATASET PREPROCESSING

This chapter explains the preprocessing techniques and machine learning utilities used before model training. Proper preprocessing ensures that the dataset is suitable for learning algorithms and improves prediction accuracy.

Libraries and Modules Used

The following Python libraries from Scikit-learn are used in this project:

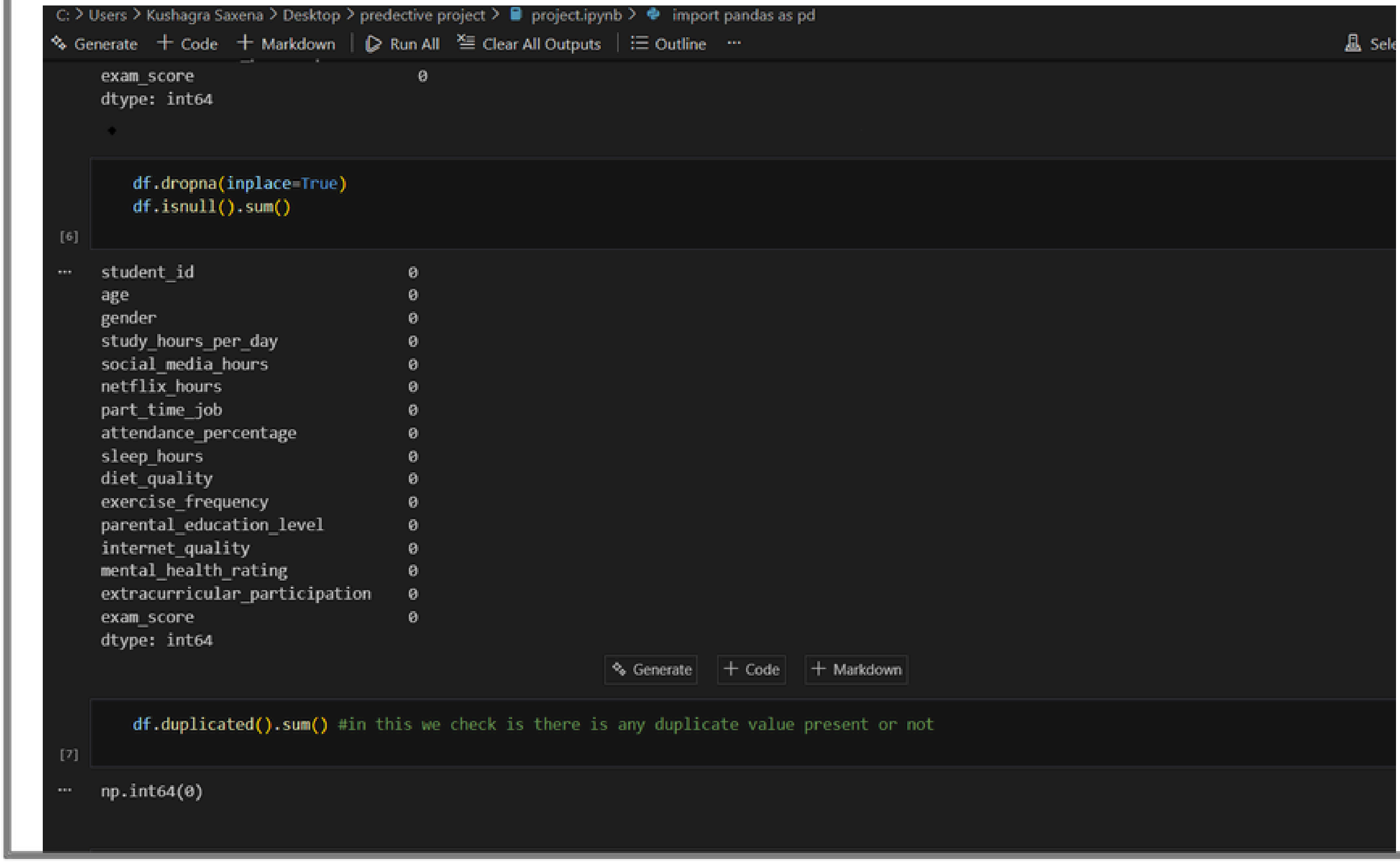
- train_test_split: Used to divide the dataset into training and testing subsets. This helps in evaluating model performance on unseen data.
- GridSearchCV: Used for hyperparameter tuning to find the optimal parameters for machine learning models.
- LabelEncoder: Converts categorical variables into numerical format so that machine learning algorithms can process them.
- mean_squared_error (MSE): Measures the average squared difference between actual and predicted values.
- r2_score (R^2): Evaluates how well the regression model explains the variance in the target variable.


C: > Users > Kushagra Saxena > Desktop > predective project > project.ipynb > import pandas as pd
Generate
+ Code
+ Markdown
Run All
Clear All Outputs
Outline
···
[3]

```python
df.info()
```

```
<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
```

```python
sns.set(style="whitegrid") #it is for better visualization
```


C: > Users > Kushagra Saxena > Desktop > predective project > project.ipynb > import pandas as pd
Generate
+ Code
+ Markdown
Run All
Clear All Outputs
Outline
···
Sele
[6]
Generate
+ Code
+ Markdown
[7]

```
exam_score                          0
dtype: int64
```

```python
df.dropna(inplace=True)
df.isnull().sum()
```

```
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         0
internet_quality                 0
mental_health_rating             0
extracurricular_participation    0
exam_score                       0
dtype: int64
```

```python
df.duplicated().sum() #in this we check is there is any duplicate value present or not
```

```
np.int64(0)
```

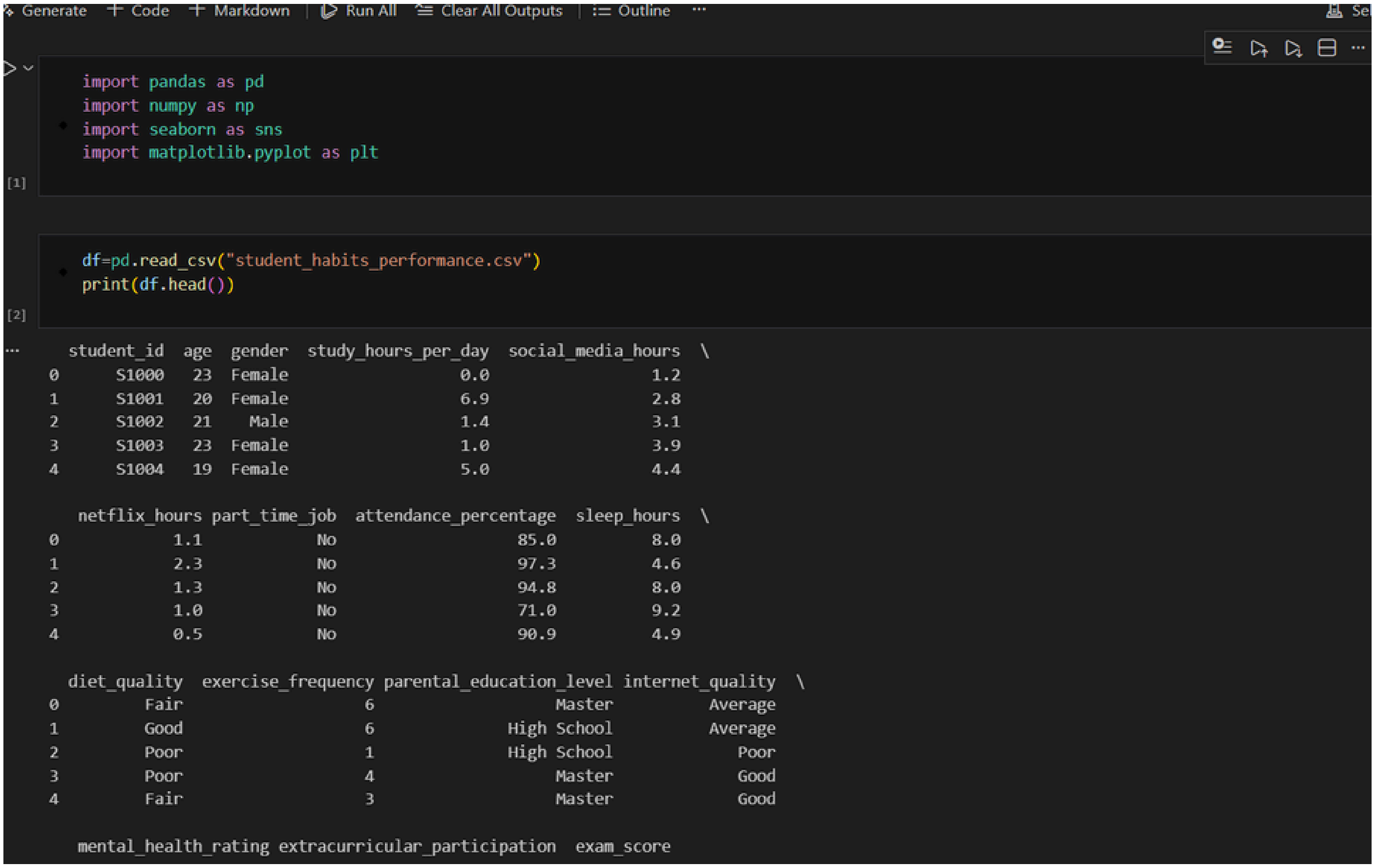
Dataset preprocessing is a crucial step to ensure data quality and improve model performance. Since the dataset is manually generated, preprocessing helps in standardizing and validating the data.

Preprocessing Steps Performed:

1. Data Cleaning:
   - Verified that there are no null or inconsistent values
   - Ensured all numeric values fall within realistic academic ranges
2. Data Formatting:
   - Converted all columns into appropriate data types
   - Renamed columns for clarity and consistency
3. Feature Selection:
   - Selected relevant independent variables affecting exam scores
   - Removed unnecessary or redundant attributes
4. Feature Scaling:
   - Applied normalization/standardization where required
   - Ensured uniform data distribution for better model learning

5. Train-Test Split:
    a. Dataset split into training and testing sets (e.g., 80% training and 20% testing)

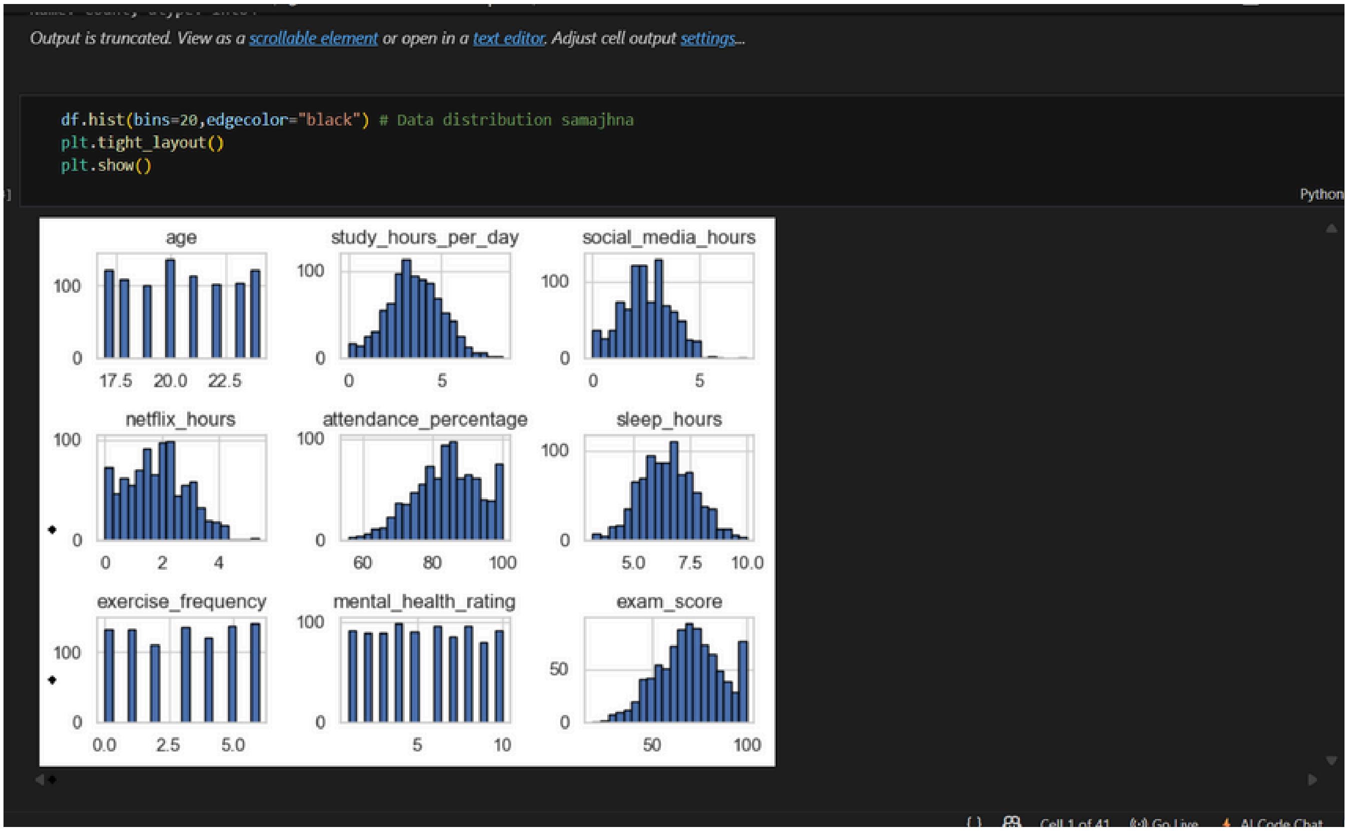
These preprocessing steps ensure reliable analysis and accurate prediction results.



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

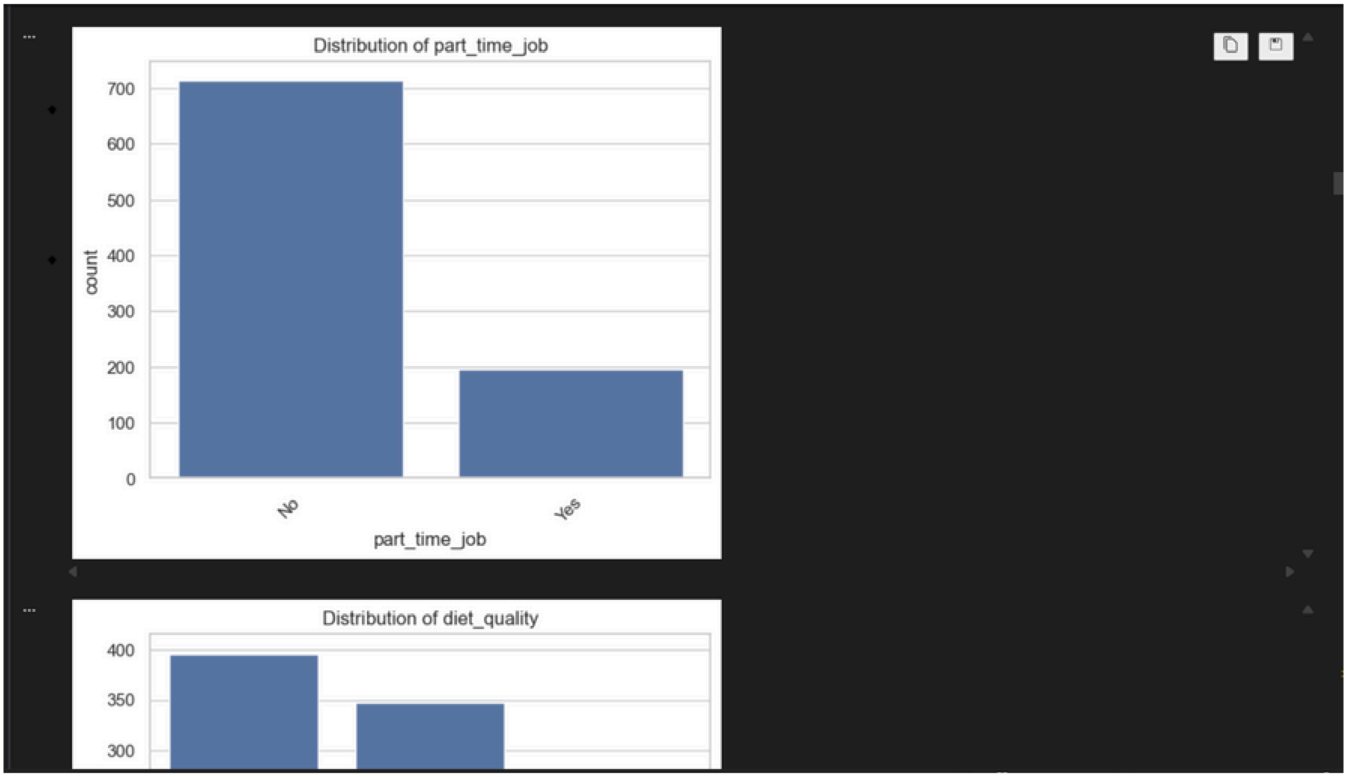
```
df=pd.read_csv("student_habits_performance.csv")
print(df.head())
```

```
  student_id  age  gender  study_hours_per_day  social_media_hours  \
0      S1000   23  Female                  0.0                 1.2
1      S1001   20  Female                  6.9                 2.8
2      S1002   21    Male                  1.4                 3.1
3      S1003   23  Female                  1.0                 3.9
4      S1004   19  Female                  5.0                 4.4

   netflix_hours part_time_job  attendance_percentage  sleep_hours  \
0            1.1            No                   85.0          8.0
1            2.3            No                   97.3          4.6
2            1.3            No                   94.8          8.0
3            1.0            No                   71.0          9.2
4            0.5            No                   90.9          4.9

  diet_quality  exercise_frequency parental_education_level internet_quality  \
0         Fair                   6                   Master          Average
1         Good                   6              High School          Average
2         Poor                   1              High School             Poor
3         Poor                   4                   Master             Good
4         Fair                   3                   Master             Good

   mental_health_rating extracurricular_participation  exam_score
```



Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
df.hist(bins=20,edgecolor="black") # Data distribution samajhna
plt.tight_layout()
plt.show()
```



4. ANALYSIS ON DATASET

This chapter presents a detailed analysis of the dataset and explains the complete machine learning workflow starting from model selection to performance evaluation. The analysis is conducted for each defined objective of the project.

MACHINE LEARNING MODELS USED AND TRAINING APPROACH

This project follows a supervised machine learning regression approach to predict student exam scores. Multiple regression models are trained and evaluated from the beginning to identify the best-performing model.

The purpose of training multiple models is to compare their performance and select the most accurate and reliable one for deployment.

TYPES OF MODELS USED

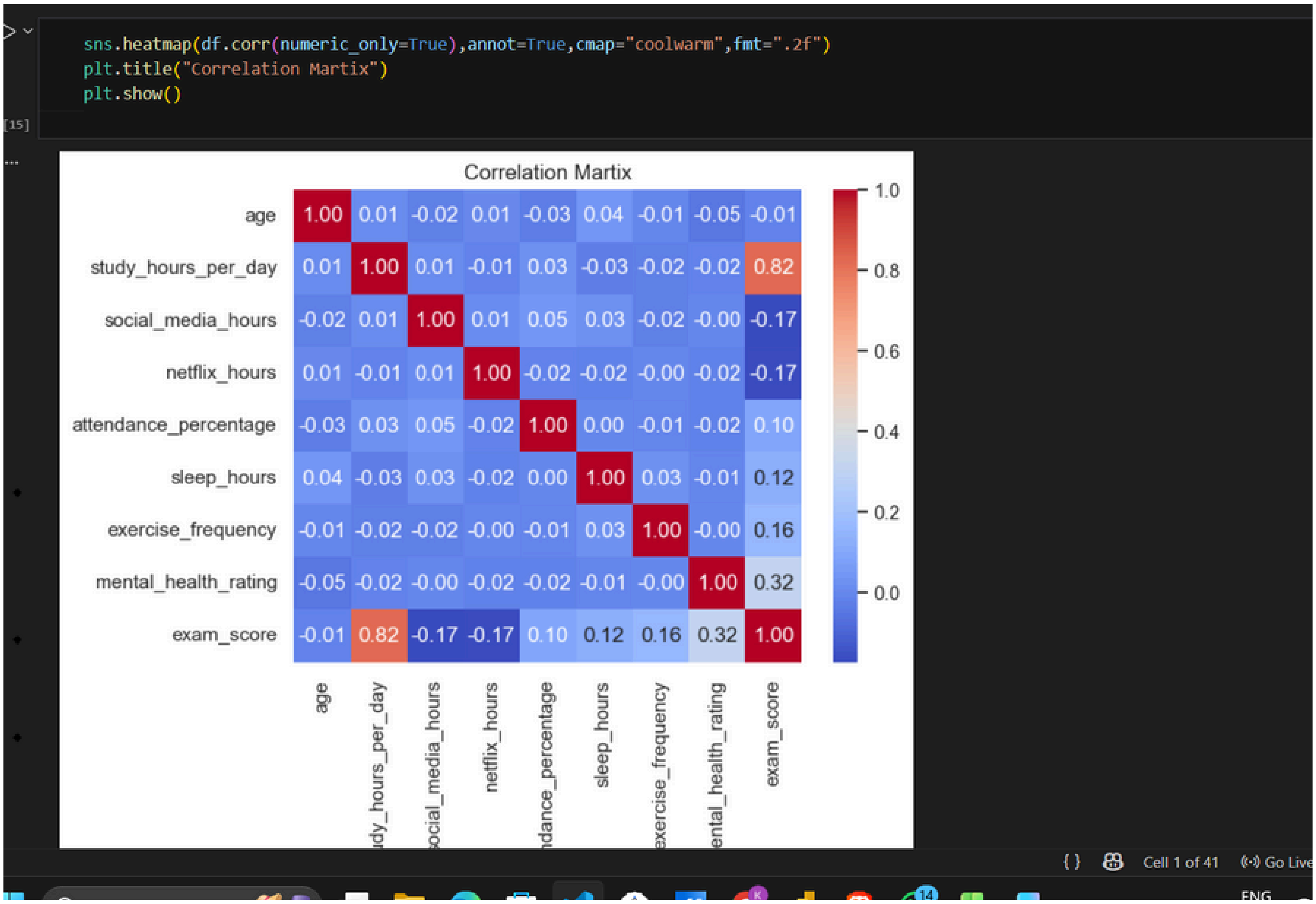
1. Linear Regression Model

Linear Regression is one of the most fundamental regression algorithms. It establishes a linear relationship between independent variables and the dependent variable (exam score).
Mathematical Representation:

$$Exam\_Score = \beta_0 + \beta_1(Study\_Hours) + \beta_2(Attendance) + \beta_3(Mental\_Health) + \beta_4(Sleep\_Hours) + \beta_5(Part\_Time\_Job) + \varepsilon$$

Why Linear Regression was used:

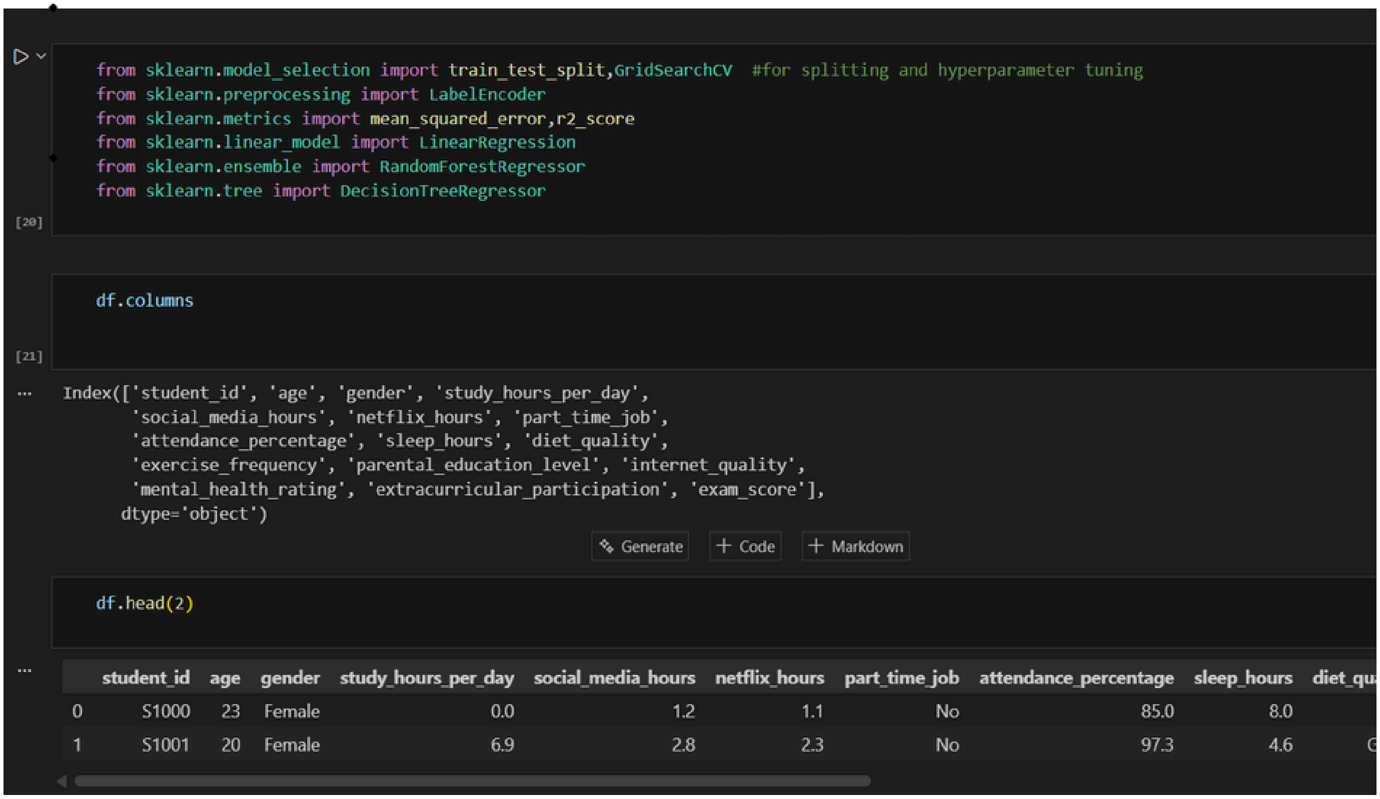
- Simple and interpretable
- Helps understand feature impact
- Serves as a baseline model



```
sns.heatmap(df.corr(numeric_only=True),annot=True,cmap="coolwarm",fmt=".2f")
plt.title("Correlation Martix")
plt.show()
```

Correlation Martix

	age	udy_hours_per_day	ocial_media_hours	netflix_hours	ndance_percentage	sleep_hours	exercise_frequency	ental_health_rating	exam_score
age	1.00	0.01	-0.02	0.01	-0.03	0.04	-0.01	-0.05	-0.01
study_hours_per_day	0.01	1.00	0.01	-0.01	0.03	-0.03	-0.02	-0.02	0.82
social_media_hours	-0.02	0.01	1.00	0.01	0.05	0.03	-0.02	-0.00	-0.17
netflix_hours	0.01	-0.01	0.01	1.00	-0.02	-0.02	-0.00	-0.02	-0.17
attendance_percentage	-0.03	0.03	0.05	-0.02	1.00	0.00	-0.01	-0.02	0.10
sleep_hours	0.04	-0.03	0.03	-0.02	0.00	1.00	0.03	-0.01	0.12
exercise_frequency	-0.01	-0.02	-0.02	-0.00	-0.01	0.03	1.00	-0.00	0.16
mental_health_rating	-0.05	-0.02	-0.00	-0.02	-0.02	-0.01	-0.00	1.00	0.32
exam_score	-0.01	0.82	-0.17	-0.17	0.10	0.12	0.16	0.32	1.00



```
from sklearn.model_selection import train_test_split,GridSearchCV  #for splitting and hyperparameter tuning
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
```

```
df.columns
```

```
Index(['student_id', 'age', 'gender', 'study_hours_per_day',
       'social_media_hours', 'netflix_hours', 'part_time_job',
       'attendance_percentage', 'sleep_hours', 'diet_quality',
       'exercise_frequency', 'parental_education_level', 'internet_quality',
       'mental_health_rating', 'extracurricular_participation', 'exam_score'],
      dtype='object')
```

```
df.head(2)
```

```
   student_id  age  gender  study_hours_per_day  social_media_hours  netflix_hours  part_time_job  attendance_percentage  sleep_hours  diet_qu
0       S1000   23  Female                  0.0                 1.2            1.1             No                   85.0          8.0
1       S1001   20  Female                  6.9                 2.8            2.3             No                   97.3          4.6        G
```

2. Random Forest Regression Model

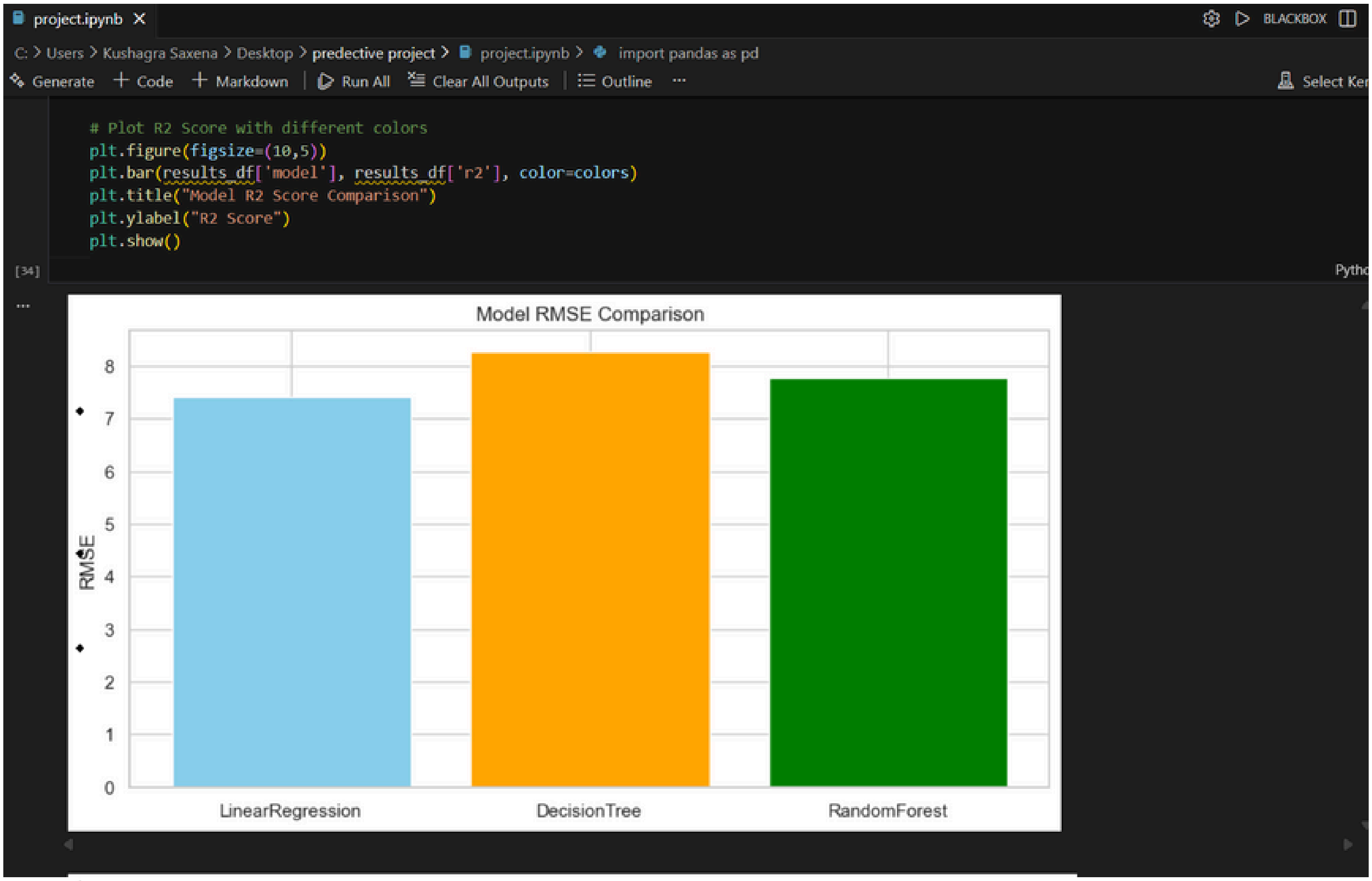
Random Forest is an ensemble learning technique that combines multiple decision trees to improve prediction accuracy and reduce overfitting.

Working Principle:

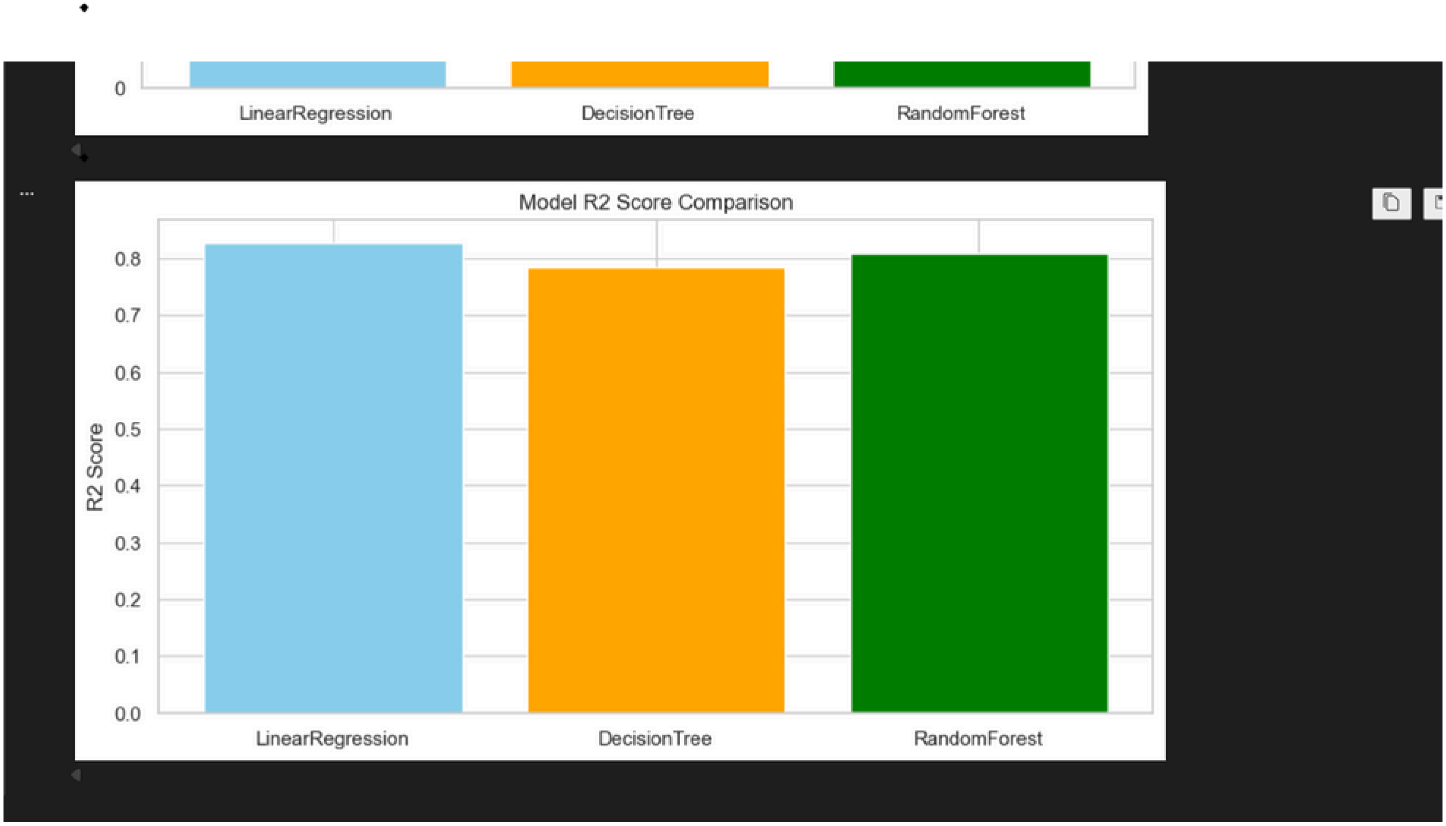
- Multiple decision trees are trained on random subsets of data
- Each tree gives a prediction
- Final output is the average of all predictions

Why Random Forest was used:

- Handles non-linear relationships effectively
- More robust than single models
- Reduces variance and improves accuracy



```
# Plot R2 Score with different colors
plt.figure(figsize=(10,5))
plt.bar(results_df['model'], results_df['r2'], color=colors)
plt.title("Model R2 Score Comparison")
plt.ylabel("R2 Score")
plt.show()
```

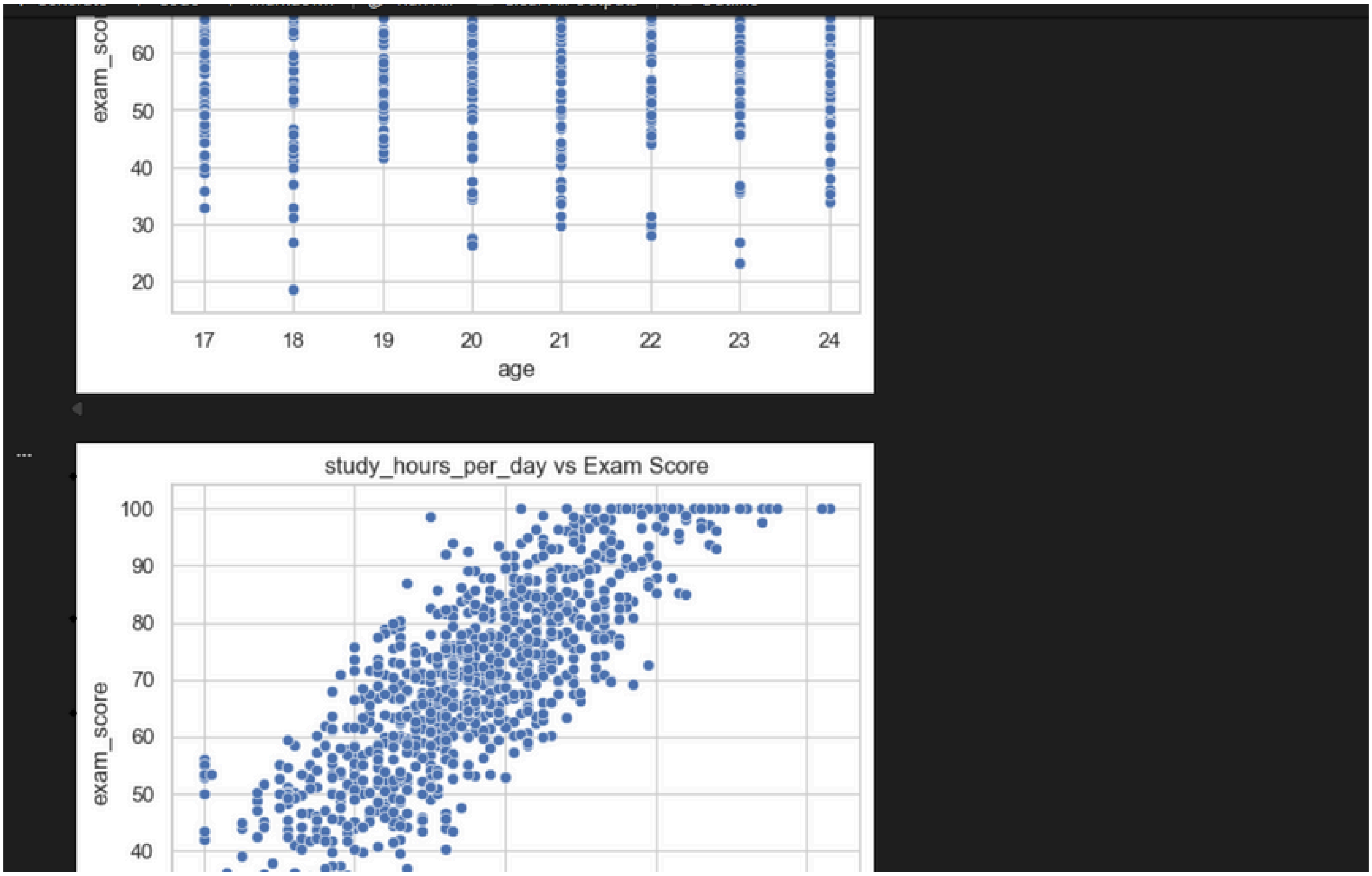


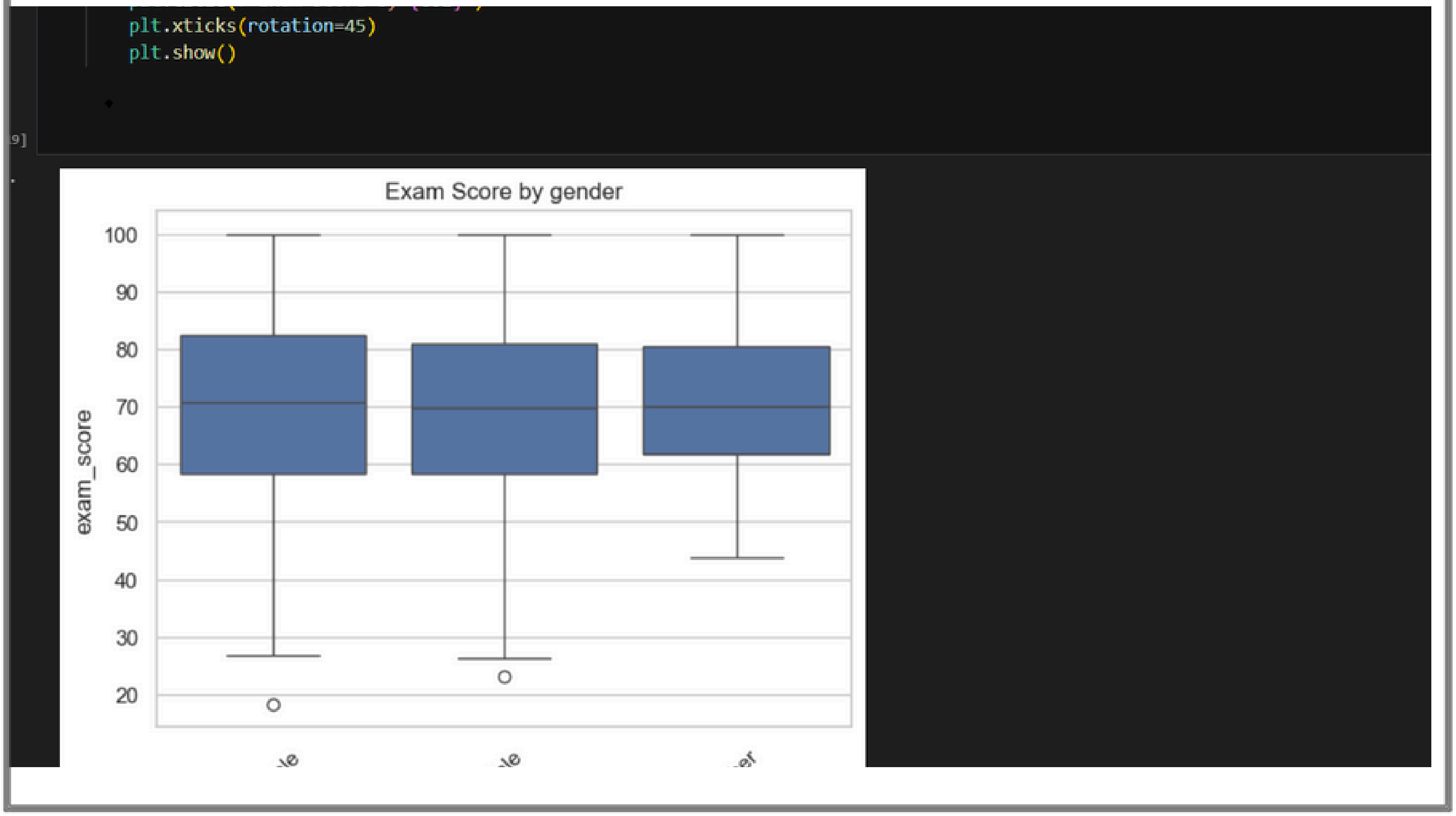
MODEL TRAINING PROCESS

The model training process is carried out in a structured and systematic manner:

1. Import required libraries and modules
2. Load the manually generated dataset
3. Encode categorical features using LabelEncoder
4. Separate independent variables (features) and dependent variable (exam score)
5. Split data into training and testing sets using train_test_split
6. Train Linear Regression model
7. Train Decision Tree Regression model
8. Train Random Forest Regression model
9. Apply GridSearchCV for hyperparameter tuning
10. Evaluate all models using MSE and R^2 score
11. Select the best-performing model
12. Save the trained model as best_model.pkl

This comparative approach ensures robustness and high predictive performance.





```
    plt.xticks(rotation=45)
    plt.show()
```

MODEL TRAINING PROCESS

The training process follows these steps:

1. Load the manually generated dataset
2. Separate input features and target variable
3. Split the dataset into training and testing sets
4. Train Linear Regression model
5. Train Random Forest Regression model
6. Evaluate both models using performance metrics
7. Select the best-performing model
8. Save the final model as best_model.pkl

The selected model is later integrated into a Streamlit-based web application for real-time prediction.

MODEL INPUT FEATURES

The following features are used as input to the model:

1. Study Hours per Day – Number of hours a student studies daily
2. Attendance Percentage – Student's class attendance
3. Mental Health Rating – Self-assessed mental health score (1–10)
4. Sleep Hours per Night – Average sleep duration
5. Part-Time Job – Whether the student has a part-time job (encoded as 0 or 1)

Target Variable:

- Final Exam Score (0–100)

These features are selected based on real-world academic research and logical assumptions affecting student performance.

i. GENERAL DESCRIPTION

The dataset consists of multiple independent variables affecting student performance. Exploratory Data Analysis (EDA) is performed to understand data distribution, trends, and correlations between variables.

ii. Specific Requirements

The specific objectives of analysis are:

- To identify key factors affecting exam scores
- To study the relationship between study hours and performance
- To evaluate the impact of attendance on exam results
- To build a prediction model with high accuracy

iii. Analysis Results

After performing Exploratory Data Analysis and model training, the following results were obtained:

- A strong positive relationship is observed between study hours and exam scores
- Students with consistent attendance perform better on average
- Previous exam performance acts as a strong predictor of final scores
- Assignment/internal assessment marks contribute significantly to overall performance

A Linear Regression model is used to predict the final exam score. The model is evaluated using performance metrics such as:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- R-squared Score

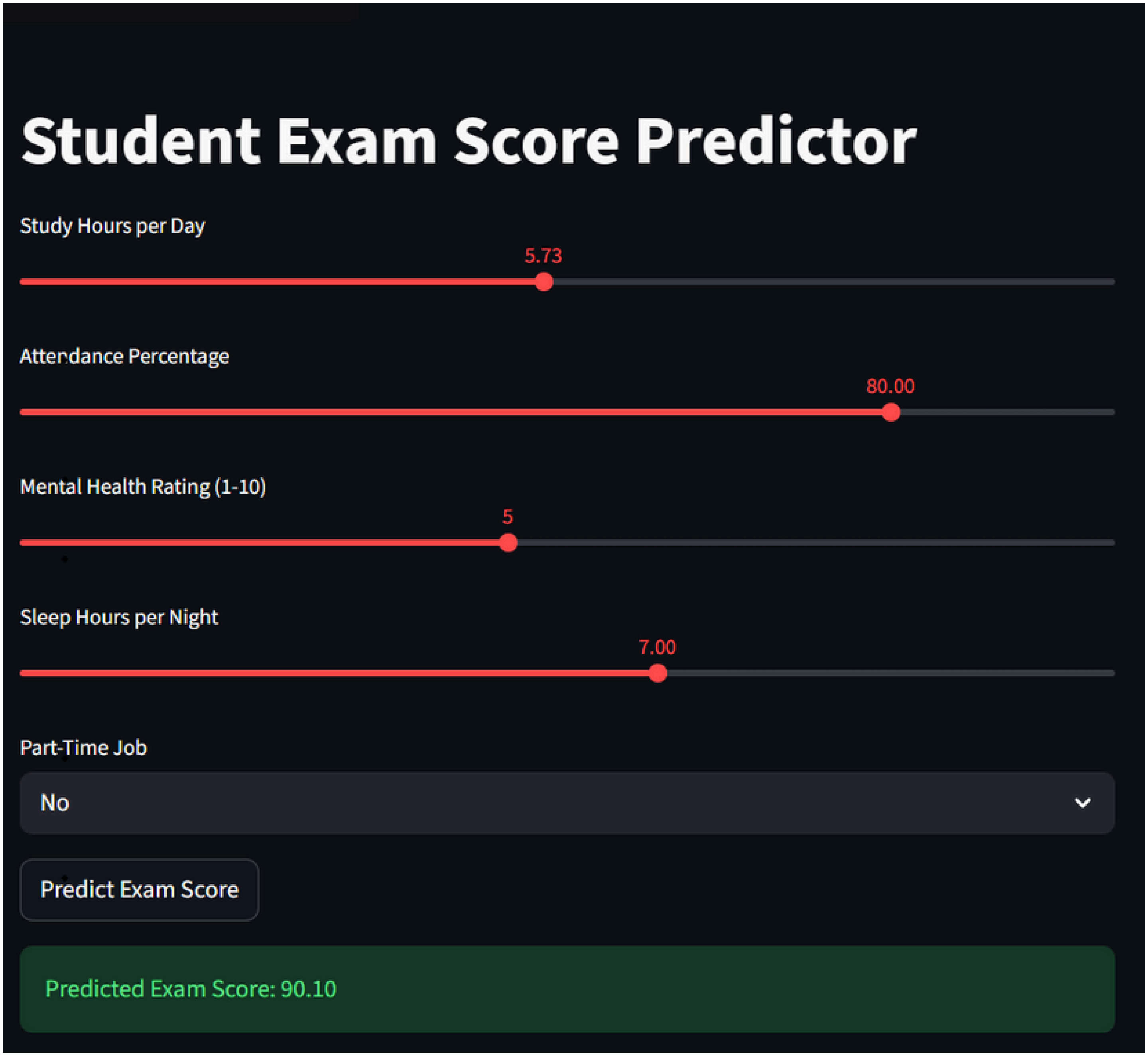
The results indicate that the model performs efficiently and provides reasonably accurate predictions on unseen data.

iv. Visualization

Data visualization is used to represent insights clearly:

- Bar charts to show score distribution
- Line graphs to analyze trends
- Scatter plots to show relationships between variables
- Heatmaps to visualize correlations

Visualization helps in understanding patterns and validating analytical results.



Student Exam Score Predictor

Study Hours per Day
5.73

Attendance Percentage
80.00

Mental Health Rating (1-10)
5

Sleep Hours per Night
7.00

Part-Time Job
No

Predicted Exam Score: 90.10

5. CONCLUSION

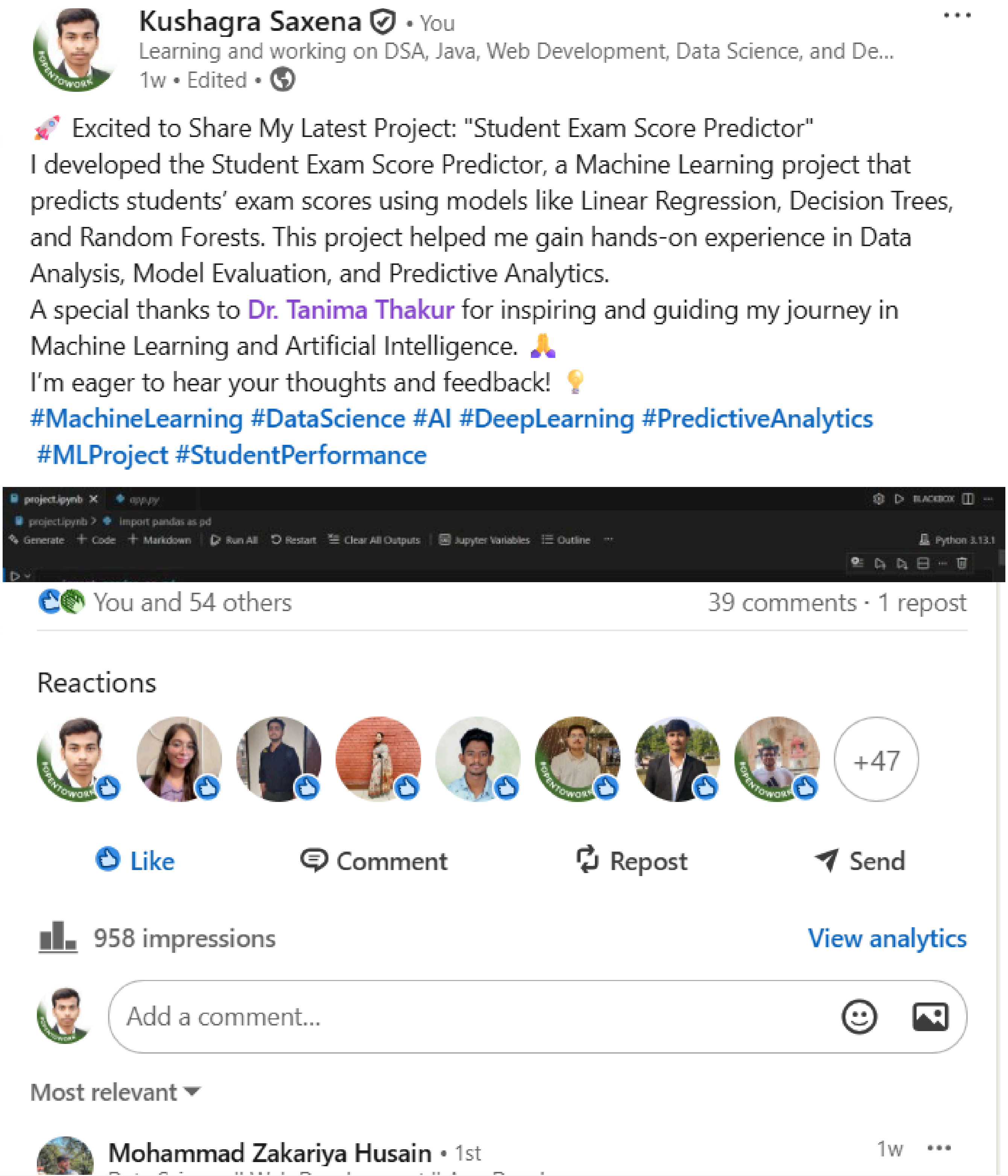
The Student Exam Score Predictor project demonstrates the effective use of data analysis and machine learning techniques in the education sector. By analyzing manually generated student data, the project successfully identifies key academic factors influencing exam performance.

The regression-based prediction model provides accurate estimates of exam scores and can be used as a decision-support tool for educators. Visualization techniques further enhance understanding by presenting insights in a clear and interpretable manner.

Overall, the project highlights the importance of data-driven approaches in improving academic outcomes and supporting student success.

LinkedIn Project URL:

https://www.linkedin.com/feed/update/urn:li:activity:7403440902695927808/



Kushagra Saxena • You

Learning and working on DSA, Java, Web Development, Data Science, and De...

1w • Edited •

🚀 Excited to Share My Latest Project: "Student Exam Score Predictor"

I developed the Student Exam Score Predictor, a Machine Learning project that predicts students' exam scores using models like Linear Regression, Decision Trees, and Random Forests. This project helped me gain hands-on experience in Data Analysis, Model Evaluation, and Predictive Analytics.

A special thanks to Dr. Tanima Thakur for inspiring and guiding my journey in Machine Learning and Artificial Intelligence. 🙏

I'm eager to hear your thoughts and feedback! 💡

#MachineLearning #DataScience #AI #DeepLearning #PredictiveAnalytics #MLProject #StudentPerformance

You and 54 others

39 comments · 1 repost

Reactions

+47

958 impressions

View analytics

Add a comment...

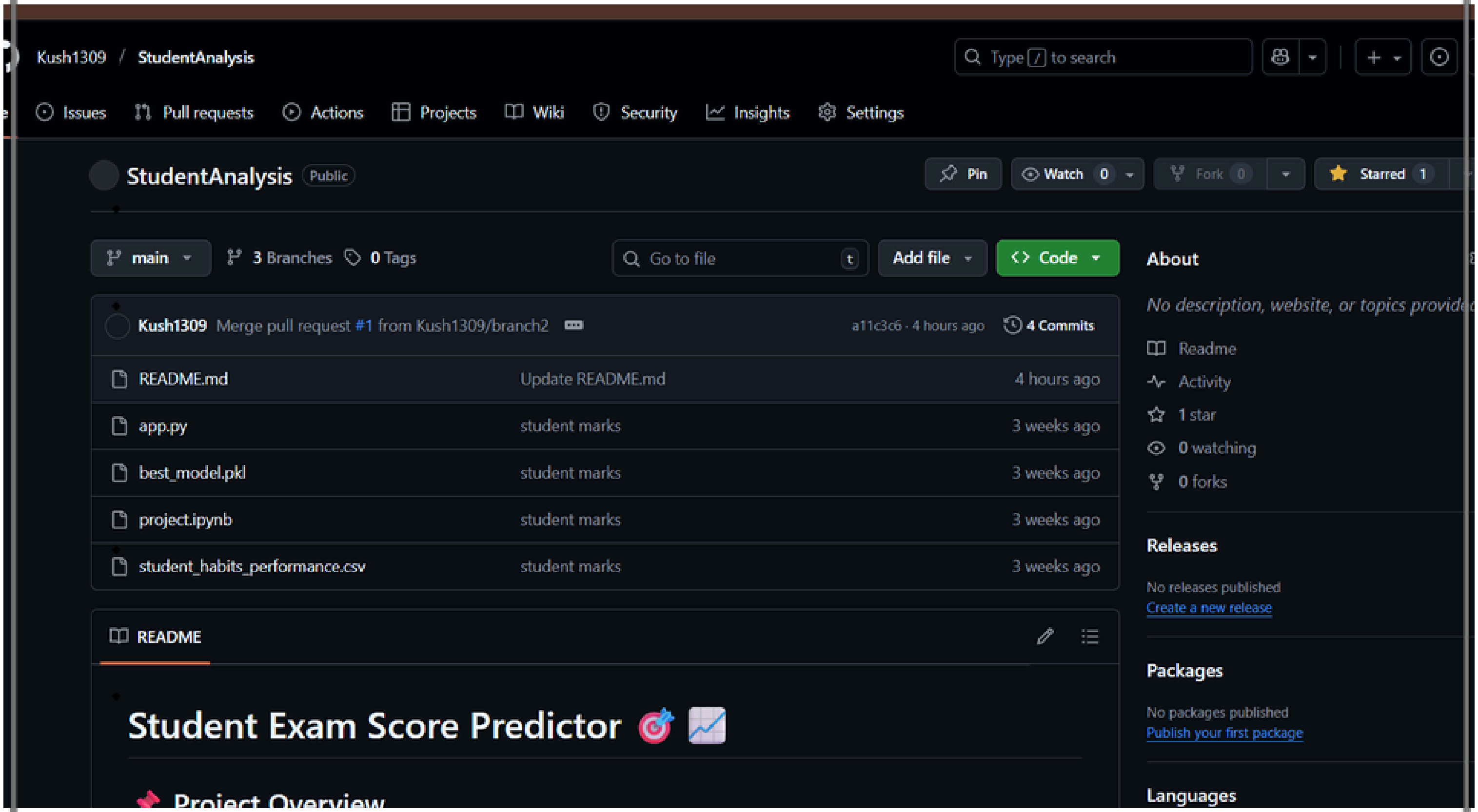
Most relevant

Mohammad Zakariya Husain • 1st

1w

GitHub Project Link:

https://github.com/Kush1309/StudentAnalysis



Kush1309 Merge pull request #1 from Kush1309/branch2 a11c3c6 · 4 hours ago 4 Commits

README.md	Update README.md	4 hours ago
app.py	student marks	3 weeks ago
best_model.pkl	student marks	3 weeks ago
project.ipynb	student marks	3 weeks ago
student_habits_performance.csv	student marks	3 weeks ago

About

No description, website, or topics provide

Readme
Activity
1 star
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

README

Student Exam Score Predictor 🎯 📈

📌 Project Overview

6. FUTURE SCOPE

The scope of the Student Exam Score Predictor project can be expanded in the following ways:

- Incorporating real-world institutional datasets for higher reliability
- Including behavioral factors such as learning patterns and online activity
- Applying advanced algorithms like Random Forest, Support Vector Machines, or Neural Networks
- Performing hyperparameter tuning to improve prediction accuracy
- Developing a web-based application for real-time score prediction
- Integrating the system with Learning Management Systems (LMS)

These enhancements can make the system more robust, scalable, and practically useful.

7. REFERENCES

- Géron, A. Hands-On Machine Learning with Scikit-Learn and TensorFlow, O’Reilly Media.
- Python Software Foundation. Python Documentation. https://docs.python.org
- Scikit-learn Developers. Scikit-learn Documentation. https://scikit-learn.org
- McKinney, W. Python for Data Analysis, O’Reilly Media.
- Kaggle. Machine Learning Tutorials and Datasets. https://www.kaggle.com
- UCI Machine Learning Repository. https://archive.ics.uci.edu