TASK 1 - Introduction to Pandas and Scikit-Learn

- This Jupyter Notebook will brief you with an introduction to Pandas and Scikit-Learn, as we get our hands dirty on a dataset.
- We will first take a look at how to view and understand the dataset, followed by some preprocessing, exploratory data analysis and visualizations of various trends in the dataset.
- We then will dive into an example of building a Machine Learning model that explores the relationship between various factors specified in the dataset and predict the 'student exam scores'.
- The aim is to understand these relationships and predict exam scores using a ML model (scikit-learn)

Student Exam Score Analysis Using Python

Objective

Analyze the student performance dataset (`student-mat.csv`) to answer specific questions using Python libraries. This project focuses on fundamental data handling, analysis, and visualization techniques.

➔ CODING.....

To complete this task, we'll follow the steps outlined and provide a comprehensive solution.

Step 1: Load the Dataset

First, we need to load the dataset using pandas. We'll use the read_csv function to load the data from the CSV file.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read_csv('student-mat.csv')

print(df.head())
```

Step 2: Data Exploration.

```
print(df.isnull().sum())
print(df.dtypes)
print(df.shape)
```

Step 3: Data Cleaning

We'll handle missing values and remove duplicates.

Replace missing values with median (if any)

```
for col in df.columns:
   if df[col].dtype == np.float64 or df[col].dtype == np.int64:
      df[col] = df[col].fillna(df[col].median())
```

Remove duplicate entries

```
df = df.drop_duplicates()
```

Step 4: Data Analysis

1.Average Score in Math (G3)

```
-> average_score = df['G3'].mean()
print(f"Average score in math (G3): {average_score}")
```

2. Number of Students Scored Above 15

```
-> students_above_15 = df[df['G3'] > 15].shape[0]
print(f"Number of students scored above 15: {students_above_15}")
```

3. Correlation Between Study Time and Final Grade

```
-> correlation = df['studytime'].corr(df['G3'])
print(f"Correlation between study time and final grade: {correlation}")
```

4. Average Final Grade by Gender

-> average_grade_by_gender = df.groupby('sex')['G3'].mean()

print(f"Average final grade by gender: \n{average_grade_by_gender}")

Step 5: Data Visualization

Let's create the required visualizations.

1. Histogram of Final Grades

-> plt.hist(df['G3'], bins=10, edgecolor='black')

plt.xlabel('Final Grade')

plt.ylabel('Frequency')

plt.title('Histogram of Final Grades')

plt.show()

2. Scatter Plot Between Study Time and Final Grade

-> plt.scatter(df['studytime'], df['G3'])

plt.xlabel('Study Time')

plt.ylabel('Final Grade')

plt.title('Scatter Plot Between Study Time and Final Grade')

plt.show()

3. Bar Chart Comparing Average Scores

->average_scores = df.groupby('sex')['G3'].mean()

average_scores.plot(kind='bar')

plt.xlabel('Gender')

plt.ylabel('Average Score')

plt.title('Average Scores by Gender’)

```
plt.show()
```