import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

Problem Step: Import all required Python libraries.

Definition:

pandas as pd: Used for handling tabular data in DataFrames.

matplotlib.pyplot as plt: Used for basic plotting (bar, line, etc.).

seaborn as sns: Statistical plotting built on top of matplotlib.

Why: These are essential for data loading (pandas) and visualization (matplotlib, seaborn) throughout the notebook.

✅ Cell 2

python

Copy

Edit

df = pd.read\_csv("StudentsPerformance.csv")

df

Problem Step: Load the dataset into memory.

Definition:

pd.read\_csv(): Reads a CSV file and creates a DataFrame.

Why: Loads the “StudentsPerformance.csv” file for analysis.

✅ Cell 3

python

Copy

Edit

df.info()

Problem Step: Check the structure of the dataset.

Definition:

.info(): Displays column names, data types, and missing value counts.

Why: Helps assess data types and detect any missing entries before analysis.

✅ Cell 4

python

Copy

Edit

df.describe()

Problem Step: Summarize numerical data.

Definition:

.describe(): Gives stats like mean, standard deviation, min, max, etc., for numeric columns.

Why: Helps understand the central tendency and variability of scores.

✅ Cell 5

python

Copy

Edit

df.isnull().sum()

Problem Step: Check for missing values.

Definition:

.isnull(): Detects null (missing) entries.

.sum(): Counts total missing values per column.

Why: Ensures dataset completeness. If any columns have missing values, they need to be handled before modeling or plotting.

✅ Cell 6

python

Copy

Edit

df.duplicated().sum()

Problem Step: Find duplicate rows.

Definition:

.duplicated(): Flags duplicated rows as True.

.sum(): Counts the total number of duplicates.

Why: Duplicate rows may distort the analysis and should be removed.

✅ Cell 7

python

Copy

Edit

df = df.drop\_duplicates()

Problem Step: Remove duplicate records.

Definition:

.drop\_duplicates(): Deletes repeated rows from the DataFrame.

Why: To avoid redundancy and ensure the accuracy of analysis.

✅ Cell 8

python

Copy

Edit

df['total score'] = df['math score'] + df['reading score'] + df['writing score']

df['average score'] = df['total score']/3

df

Problem Step: Create new metrics for total and average scores.

Definition:

df['total score']: Sum of the three subject scores.

df['average score']: Mean score across all three subjects.

Why: Helps in understanding overall performance and comparing students holistically.

✅ Cell 9

python

Copy

Edit

sns.histplot(df['average score'], kde=True)

plt.title("Distribution of Average Scores")

plt.xlabel("Average Score")

plt.ylabel("Frequency")

plt.show()

Problem Step: Visualize how average scores are distributed.

Definitions:

sns.histplot(): Plots a histogram; kde=True adds a smooth curve showing distribution.

plt.title/xlabel/ylabel(): Sets chart labels.

Why: To understand if student performance is normally distributed or skewed.

✅ Cell 10

python

Copy

Edit

sns.boxplot(x='gender', y='average score', data=df)

plt.title("Average Score Based on Gender")

plt.show()

Problem Step: Compare average scores by gender.

Definition:

sns.boxplot(): Visual summary of score distribution with median, quartiles, and outliers.

Why: Helps identify performance differences between male and female students.

✅ Cell 11

python

Copy

Edit

sns.barplot(x='test preparation course', y='average score', data=df)

plt.title("Average Score Based on Test Preparation")

plt.show()

Problem Step: Analyze how test prep impacts scores.

Definition:

sns.barplot(): Shows average score per category using bars.

Why: To evaluate the effectiveness of test preparation courses.

✅ Cell 12

python

Copy

Edit

sns.violinplot(x='parental level of education', y='average score', data=df)

plt.title("Average Score Based on Parental Education Level")

plt.xticks(rotation=45)

plt.show()

Problem Step: Assess influence of parental education.

Definition:

sns.violinplot(): Combines boxplot and KDE to show score distributions.

plt.xticks(rotation=45): Rotates x-axis labels for readability.

Why: Parental education can affect academic support at home, influencing student scores.

✅ Cell 13

python

Copy

Edit

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title("Correlation Matrix")

plt.show()

Problem Step: Examine correlation between variables.

Definitions:

df.corr(): Computes pairwise correlation.

sns.heatmap(): Displays the correlation matrix.

Why: Useful for detecting multicollinearity or strong linear relationships.

✅ Cell 14

python

Copy

Edit

plt.figure(figsize=(12,6))

sns.countplot(x='race/ethnicity', hue='gender', data=df)

plt.title("Race/Ethnicity Distribution by Gender")

plt.show()

Problem Step: Understand gender representation within ethnic groups.

Definition:

sns.countplot(): Bar chart of counts for each category.

hue='gender': Splits bars by gender.

Why: Helps explore if gender and race distribution are balanced.