

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

Choose Files | No file chosen
this cell to enable.

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving Stud.csv to Stud (?) csv

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

```
df=pd.read_csv('Stud.csv')
df.head(10)
```

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score
0	female	group B	bachelor's degree	standard	none	72	72	
1	female	group C	some college	standard	completed	69	90	
2	female	group B	master's degree	standard	none	90	95	
3	male	group A	associate's degree	free/reduced	none	47	57	
4	male	group C	some college	standard	none	76	78	
5	female	group B	associate's degree	standard	none	71	83	
6	female	group B	some college	standard	completed	88	95	
7	male	group B	some college	free/reduced	none	40	43	
8	male	group D	high school	free/reduced	completed	64	64	
9	female	group B	high school	free/reduced	none	38	60	

```
student_data=pd.read_csv("Stud.csv")
print(student_data.head())
```

	gender	race_ethnicity	parental_level_of_education	lunch
0	female	group B	bachelor's degree	standard
1	female	group C	some college	standard
2	female	group B	master's degree	standard
3	male	group A	associate's degree	free/reduced
4	male	group C	some college	standard

	test_preparation_course	math_score	reading_score	writing_score
0	none	72	72	74
1	completed	69	90	88
2	none	90	95	93
3	none	47	57	44
4	none	76	78	75

student_data.shape

(1000, 8)

student_data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   gender          1000 non-null    object  
 1   race_ethnicity  1000 non-null    object  
 2   parental_level_of_education 1000 non-null    object  
 3   lunch            1000 non-null    object  
 4   test_preparation_course 1000 non-null    object  
 5   math_score       1000 non-null    int64  
 6   reading_score    1000 non-null    int64  
 7   writing_score    1000 non-null    int64  
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

```
student_data['gender'] = student_data['gender'].fillna(student_data['gender'].mode()[0])
print(student_data.head())
```

	gender	race_ethnicity	parental_level_of_education	lunch
0	female	group B	bachelor's degree	standard

```

1 female      group C           some college    standard
2 female      group B           master's degree standard
3 male        group A          associate's degree free/reduced
4 male        group C           some college    standard

 test_preparation_course  math_score  reading_score  writting_score
0             none            72            72            74
1            completed         69            90            88
2             none            90            95            93
3             none            47            57            44
4             none            76            78            75

```

```
student_data['lunch'].str.upper()
```

	lunch
0	STANDARD
1	STANDARD
2	STANDARD
3	FREE/REDUCED
4	STANDARD
...	...
995	STANDARD
996	FREE/REDUCED
997	FREE/REDUCED
998	STANDARD
999	FREE/REDUCED

1000 rows × 1 columns

dtype: object

```
student_data.describe()
```

	math_score	reading_score	writting_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

```
student_data.isnull().sum()
```

	0
gender	0
race_ethnicity	0
parental_level_of_education	0
lunch	0
test_preparation_course	0
math_score	0
reading_score	0
writting_score	0

dtype: int64

```
student_data.groupby("gender")["math_score"].sum()
```

```
math_score  
gender  
-----  
female    32962  
male      33127  
  
dtype: int64
```

```
student_data.groupby("gender")["math_score"].mean()
```

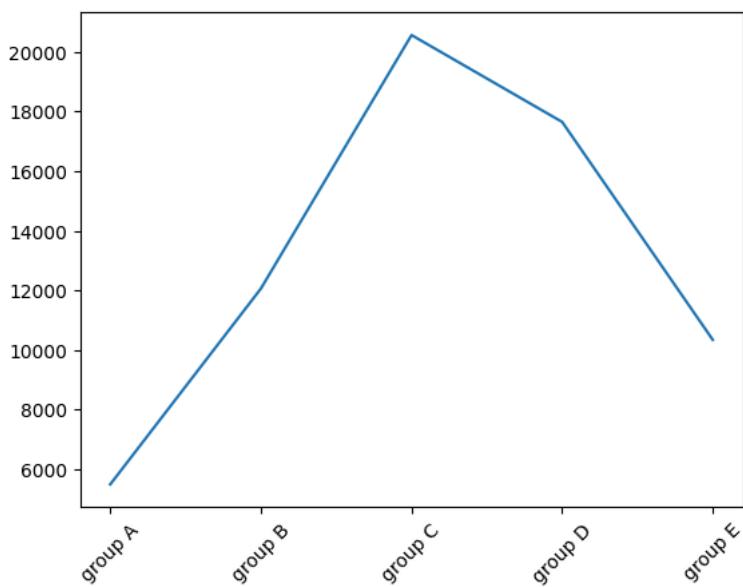
```
math_score  
gender  
-----  
female    63.633205  
male      68.728216  
  
dtype: float64
```

```
student_data["test_preparation_course"].value_counts()
```

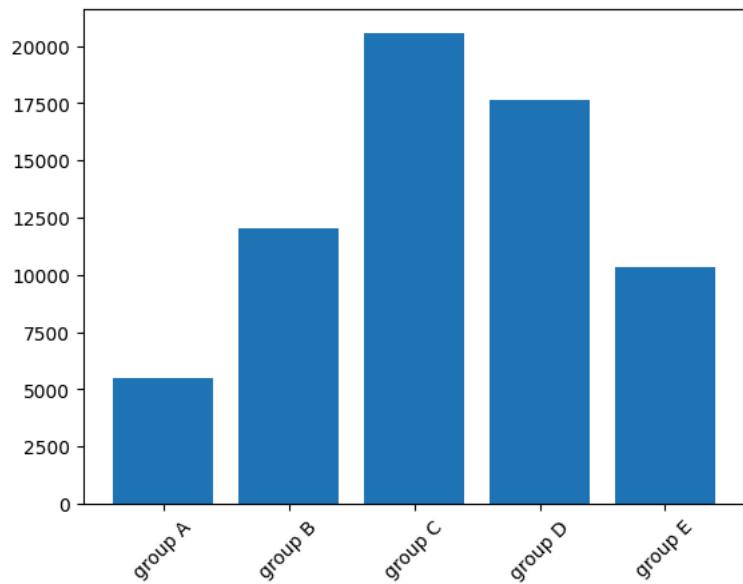
```
count  
test_preparation_course  
-----  
none        642  
completed   358  
  
dtype: int64
```

```
testing_level=student_data.groupby("race_ethnicity")["math_score"].sum()
```

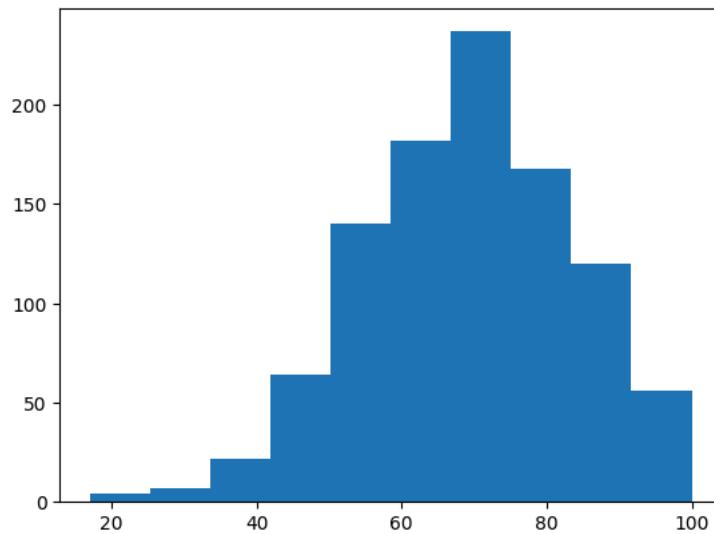
```
plt.plot(testing_level.index,testing_level.values)  
plt.xticks(rotation=45)  
plt.show()
```



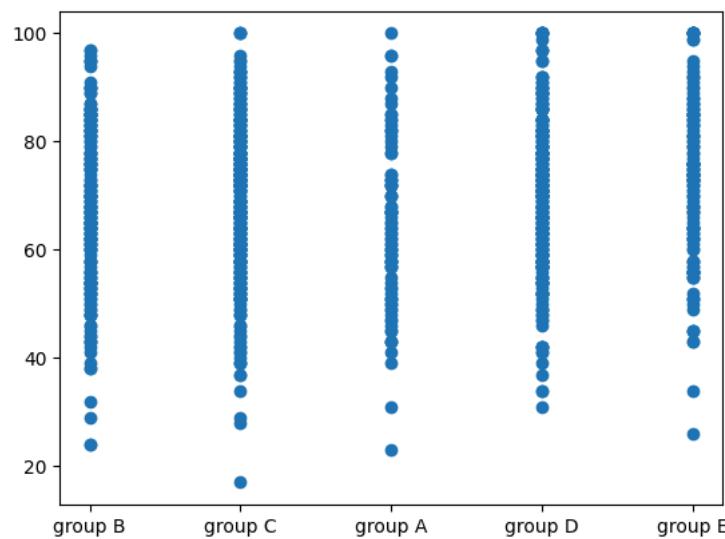
```
plt.bar(testing_level.index,testing_level.values)  
plt.xticks(rotation=45)  
plt.show()
```



```
plt.hist(student_data["reading_score"])
plt.show()
```

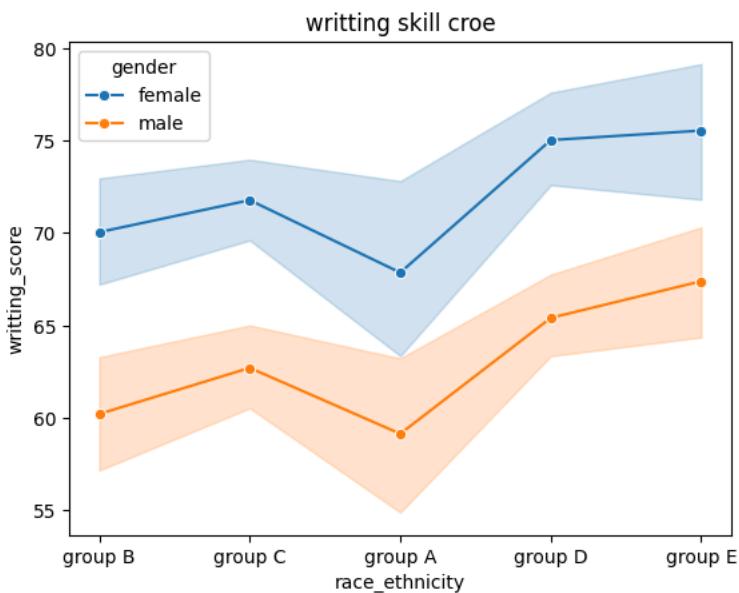


```
plt.scatter(student_data["race_ethnicity"],student_data["reading_score"])
plt.show()
```

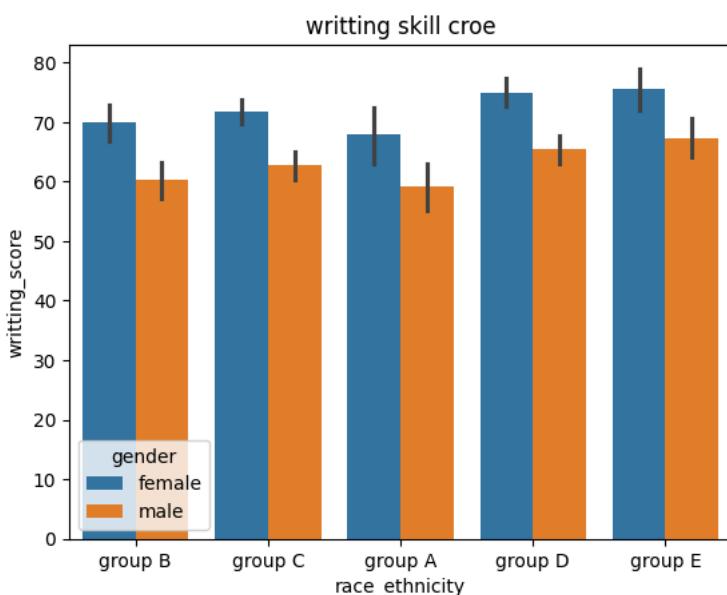


```
sns.set_theme(style="darkgrid")
```

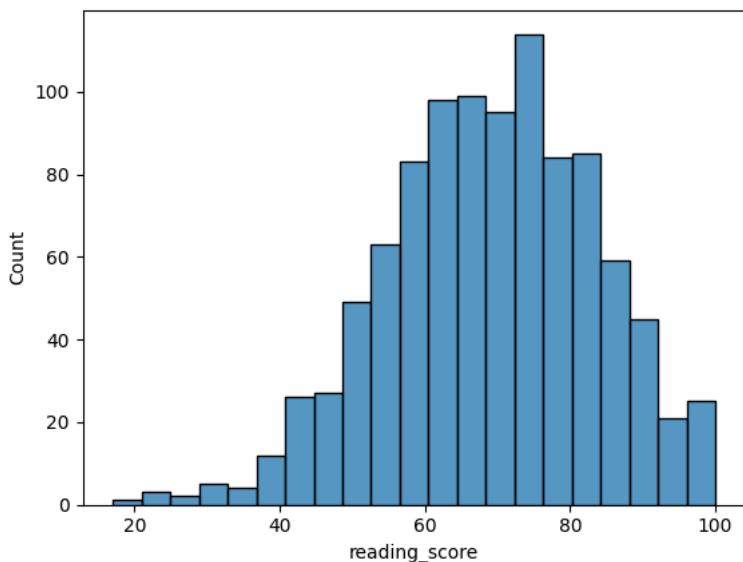
```
sns.lineplot(x='race_ethnicity',y='writting_score',hue='gender',data=student_data,marker='o')
plt.title('writting skill croe')
plt.show()
```



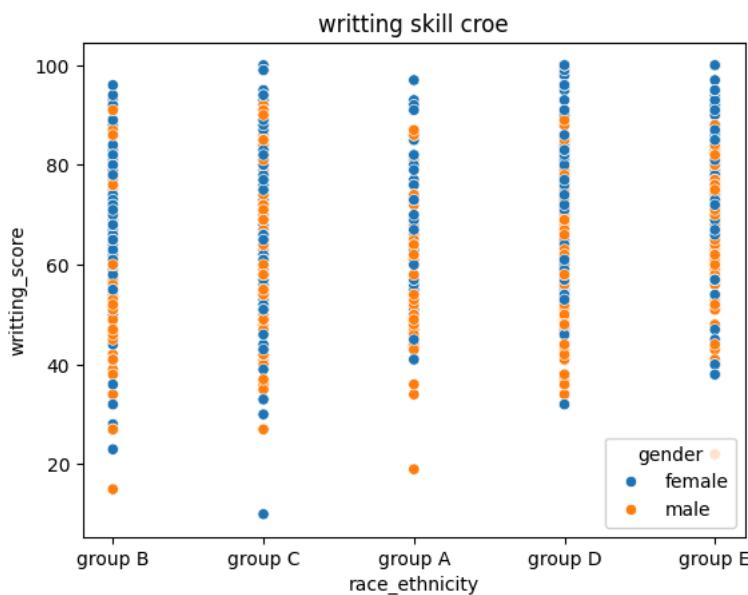
```
sns.barplot(x='race_ethnicity',y='writting_score',hue='gender',data=student_data)
plt.title('writting skill croe')
plt.show()
```



```
sns.histplot(student_data["reading_score"])
plt.show()
```



```
sns.scatterplot(x='race_ethnicity',y='writting_score',hue='gender',data=student_data)
plt.title('writting skill croe')
plt.show()
```



```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import pickle

# Load dataset
df = pd.read_csv("Stud.csv")

# Split features & target
X = df.drop("math_score", axis=1)
y = df["math_score"]

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Preprocessing
categorical_features = [
    "gender", "race_ethnicity",
    "parental_level_of_education",
    "lunch", "test_preparation_course"
]
numerical_features = ["reading_score", "writting_score"]
```

```

preprocessor = ColumnTransformer(
    transformers=[
        ("cat", OneHotEncoder(drop="first"), categorical_features),
        ("num", "passthrough", numerical_features)
    ]
)

# Model
model = Pipeline(steps=[
    ("preprocessor", preprocessor),
    ("regressor", LinearRegression())
])

# Train model
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluation
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r2)

# Save the trained model
with open("student_math_model.pkl", "wb") as f:
    pickle.dump(model, f)

```

MAE: 4.214763142474852
MSE: 29.09516986671551
RMSE: 5.393993869732845
R2 Score: 0.8804332983749564

```
!pip install gradio pandas numpy scikit-learn
```

```

Requirement already satisfied: gradio in /usr/local/lib/python3.12/dist-packages (5.50.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (2.0.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (1.6.1)
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (24.1.0)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (4.12.1)
Requirement already satisfied: brotli<=1.1.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (1.2.0)
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.123.10)
Requirement already satisfied: ffmpeg in /usr/local/lib/python3.12/dist-packages (from gradio) (1.0.0)
Requirement already satisfied: gradio-client==1.14.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (1.14.0)
Requirement already satisfied: groovy~0.1 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.1.2)
Requirement already satisfied: httpx<1.0,>=0.24.1 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.28.1)
Requirement already satisfied: huggingface-hub<2.0,>=0.33.5 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.36)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (3.0.3)
Requirement already satisfied: orjson~3.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (3.11.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from gradio) (25.0)
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (11.3.0)
Requirement already satisfied: pydantic<=2.12.3,>=2.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (2.12.3)
Requirement already satisfied: pydub in /usr/local/lib/python3.12/dist-packages (from gradio) (0.25.1)
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.0.21)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (6.0.3)
Requirement already satisfied: ruff>0.9.3 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.14.11)
Requirement already satisfied: safethttx<0.2.0,>=0.1.6 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.1.7)
Requirement already satisfied: semantic-version~2.0.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (2.10.0)
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.50.0)
Requirement already satisfied: tomllib<0.14.0,>=0.12.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.13.3)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.21.1)
Requirement already satisfied: typing-extensions<=4.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (4.15.0)
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.12/dist-packages (from gradio) (0.40.0)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from gradio-client==1.14.0->gradio) (202
Requirement already satisfied: websockets<16.0,>=13.0 in /usr/local/lib/python3.12/dist-packages (from gradio-client==1.14
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.3)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.16.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.5.3)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.12/dist-packages (from anyio<5.0,>=3.0->gradio) (3.11)
Requirement already satisfied: annotated-doc>=0.0.2 in /usr/local/lib/python3.12/dist-packages (from fastapi<1.0,>=0.115.2
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from https<1.0,>=0.24.1->gradio) (2026.
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1.0,>=0.24.1->gradio)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1.0,>=0.24.
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=0.33.5->gra
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=0.33.5->gra
```

```
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=0.33.5)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<=2.12.3,>=)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic<=2.12.3,>=)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic<=2.12.3,>=)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (from requests->huggingface-hub<2.0,>=0.33.5)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer<1.0,>=0.12->gradio) (from requests->huggingface-hub<2.0,>=0.33.5)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from typer<1.0,>=0.12->gradio) (from requests->huggingface-hub<2.0,>=0.33.5)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.12/dist-packages (from typer<1.0,>=0.12->gradio) (from requests->huggingface-hub<2.0,>=0.33.5)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->typer) (from requests->huggingface-hub<2.0,>=0.33.5)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->typer) (from requests->huggingface-hub<2.0,>=0.33.5)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->huggingface-hub<2.0,>=0.33.5)
```

```
import gradio as gr
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression

# Load dataset
df = pd.read_csv("Stud.csv")

# Features & target
X = df.drop("math_score", axis=1)
y = df["math_score"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Preprocessing
categorical_features = [
    "gender", "race_ethnicity",
    "parental_level_of_education",
    "lunch", "test_preparation_course"
]
numerical_features = ["reading_score", "writting_score"]

preprocessor = ColumnTransformer(
    transformers=[
        ("cat", OneHotEncoder(drop="first"), categorical_features),
        ("num", "passthrough", numerical_features)
    ]
)

# Model pipeline
model = Pipeline(steps=[
    ("preprocessor", preprocessor),
    ("regressor", LinearRegression())
])

# Train model (NO PICKLE)
model.fit(X_train, y_train)

# Prediction function for Gradio
def predict_math_score(
    gender,
    race_ethnicity,
    parental_level_of_education,
    lunch,
    test_preparation_course,
    reading_score,
    writting_score
):
    input_data = pd.DataFrame([
        {
            "gender": gender,
            "race_ethnicity": race_ethnicity,
            "parental_level_of_education": parental_level_of_education,
            "lunch": lunch,
            "test_preparation_course": test_preparation_course,
            "reading_score": reading_score,
            "writting_score": writting_score
        }
    ])

    prediction = model.predict(input_data)
    return round(prediction[0], 2)

# Gradio Interface
```

```
interface = gr.Interface(
    fn=predict_math_score,
    inputs=[
        gr.Dropdown(["female", "male"], label="Gender"),
        gr.Dropdown(["group A", "group B", "group C", "group D", "group E"], label="Race/Ethnicity"),
        gr.Dropdown([
            "some high school", "high school",
            "some college", "associate's degree",
            "bachelor's degree", "master's degree"
        ], label="Parental Education"),
        gr.Dropdown(["standard", "free/reduced"], label="Lunch Type"),
        gr.Dropdown(["none", "completed"], label="Test Preparation"),
        gr.Slider(0, 100, value=60, label="Reading Score"),
        gr.Slider(0, 100, value=60, label="Writing Score")
    ],
    outputs=gr.Number(label="Predicted Math Score"),
    title="🎓 Student Math Score Prediction",
    description="Gradio App deployed in Google Colab (No Pickle Used)"
)

interface.launch(share=True)
```

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: <https://44b422497af859855a.gradio.live>

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the

🎓 Student Math Score Prediction

Gradio App deployed in Google Colab (No Pickle Used)

The screenshot shows the deployed Gradio application. On the left, there are two dropdown menus: 'Gender' (set to 'female') and 'Race/Ethnicity' (set to 'group A'). On the right, there is a large input field labeled 'Predicted Math Score' containing the value '0'. Below this input field is a small button labeled 'Flag'.