```
# 🧩 Students Performance Dataset Preprocessing
# NIKHIL SHIRSATHE 202401110003

# --- Import Libraries ---
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectKBest, chi2
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# --- Load Dataset ---
# 📂 Make sure you have uploaded StudentsPerformance.csv in Colab
from google.colab import files
uploaded = files.upload()

data = pd.read_csv('StudentsPerformance.csv')
print("✅ Dataset Loaded Successfully!")
data.head()
```

Choose Files StudentsPerformance.csv

StudentsPerformance.csv(text/csv) - 72036 bytes, last modified: 11/2/2025 - 100% done

```
Saving StudentsPerformance.csv to StudentsPerformance (3).csv
✅ Dataset Loaded Successfully!
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading 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

Next steps: Generate code with `data` New interactive sheet

```
# 1️⃣ Data Exploration
print("\n--- Data Info ---")
data.info()
print("\n--- Data Description ---")
print(data.describe(include='all'))
```

```
print("\n--- Missing Values ---")
print(data.isnull().sum())
```

```
--- 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

--- Data Description ---
        gender race/ethnicity parental level of education     lunch  \
count     1000           1000                        1000      1000
unique       2              5                           6         2
top     female        group C                some college  standard
freq       518            319                         226       645
mean       NaN            NaN                         NaN       NaN
std        NaN            NaN                         NaN       NaN
min        NaN            NaN                         NaN       NaN
25%        NaN            NaN                         NaN       NaN
50%        NaN            NaN                         NaN       NaN
75%        NaN            NaN                         NaN       NaN
max        NaN            NaN                         NaN       NaN

       test preparation course  math score  reading score  writing score
count                     1000  1000.00000    1000.000000    1000.000000
unique                       2         NaN            NaN            NaN
top                       none         NaN            NaN            NaN
freq                       642         NaN            NaN            NaN
mean                       NaN    66.08900      69.169000      68.054000
std                        NaN    15.16308      14.600192      15.195657
min                        NaN     0.00000      17.000000      10.000000
25%                        NaN    57.00000      59.000000      57.750000
50%                        NaN    66.00000      70.000000      69.000000
75%                        NaN    77.00000      79.000000      79.000000
max                        NaN   100.00000     100.000000     100.000000

--- Missing Values ---
gender                         0
race/ethnicity                 0
parental level of education    0
lunch                          0
test preparation course        0
math score                     0
reading score                  0
writing score                  0
dtype: int64
```

```
# 2️⃣ Handle Missing Data
for col in data.columns:
    if data[col].isnull().sum() > 0:
        if data[col].dtype in ['float64', 'int64']:
            data[col] = data[col].fillna(data[col].median())
        else:
            data[col] = data[col].fillna(data[col].mode()[0])
print("\n✅ Missing values after handling:\n", data.isnull().sum())
```

```
✅ Missing values after handling:
 gender                         0
race/ethnicity                  0
parental level of education     0
lunch                           0
test preparation course         0
math score                      0
reading score                   0
writing score                   0
dtype: int64
```

```
# 3️⃣ Encode Categorical Variables
cat_cols = data.select_dtypes(include=['object']).columns
le = LabelEncoder()
for col in cat_cols:
    data[col] = le.fit_transform(data[col])
data = pd.get_dummies(data, columns=cat_cols, drop_first=True)
print("\n✅ Categorical variables encoded.")
data.head()
```

```
✅ Categorical variables encoded.
```

	math score	reading score	writing score	gender_1	race/ethnicity_1	race/ethnicity_2	rac
0	72	72	74	False	True	False	
1	69	90	88	False	False	True	
2	90	95	93	False	True	False	
3	47	57	44	True	False	False	
4	76	78	75	True	False	True	

Next steps: Generate code with `data` New interactive sheet

```
# 4️⃣ Feature Scaling
scaler = StandardScaler()
num_features = data.select_dtypes(include=['float64', 'int64']).columns
scaled_data = data.copy()
scaled_data[num_features] = scaler.fit_transform(data[num_features])
```

```
print("\n✅ Feature scaling complete.")
scaled_data[num_features].head()
```

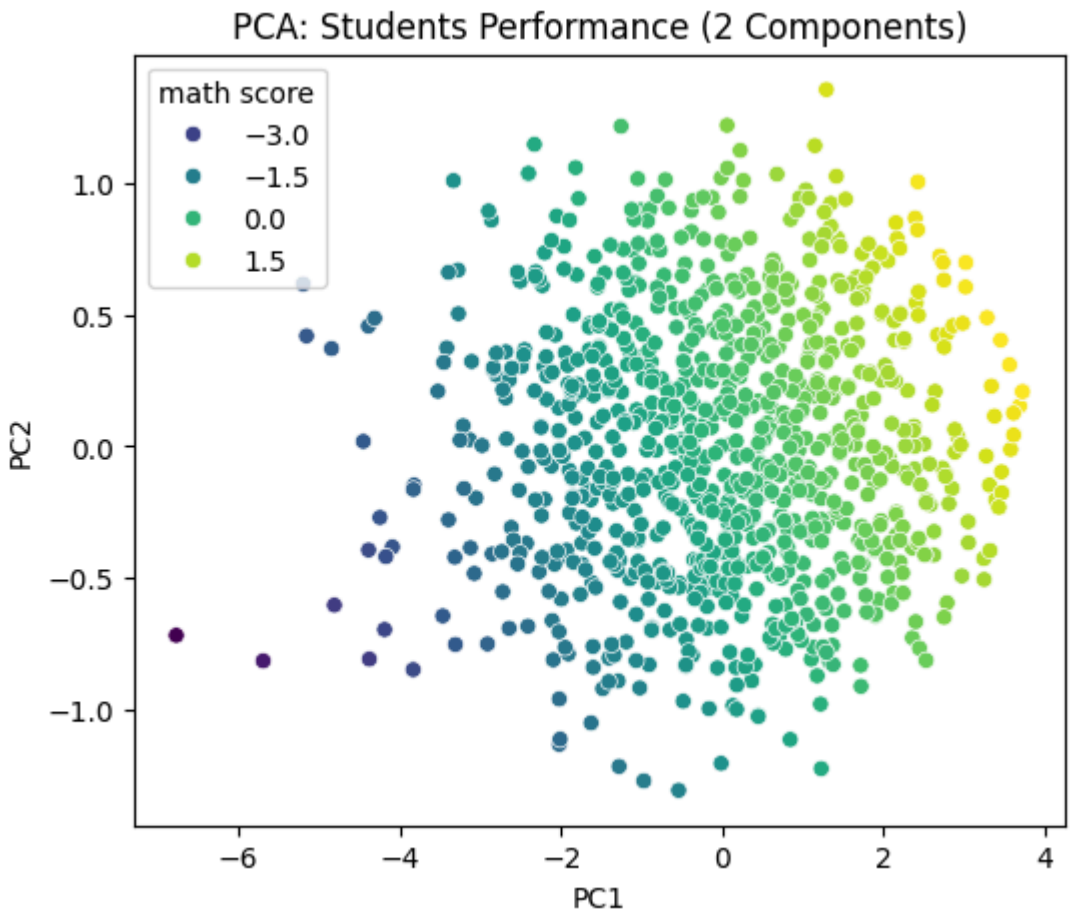
✅ Feature scaling complete.

	math score	reading score	writing score
0	0.390024	0.193999	0.391492
1	0.192076	1.427476	1.313269
2	1.577711	1.770109	1.642475
3	-1.259543	-0.833899	-1.583744
4	0.653954	0.605158	0.457333

```
# 5️⃣ PCA (Dimensionality Reduction)
numeric_data = scaled_data.select_dtypes(include=['float64', 'int64'])
pca = PCA(n_components=2)
pca_features = pca.fit_transform(numeric_data)
pca_df = pd.DataFrame(pca_features, columns=['PC1', 'PC2'])
print("\n✅ PCA completed. Sample data:")
print(pca_df.head())
```

```
✅ PCA completed. Sample data:
        PC1       PC2
0  0.560514  0.088285
1  1.719201 -0.910745
2  2.883135 -0.021999
3 -2.119921 -0.074994
4  0.988094  0.131914
```

```
# Visualization
plt.figure(figsize=(6,5))
sns.scatterplot(x='PC1', y='PC2', hue=scaled_data['math score'], palette='vi
plt.title('PCA: Students Performance (2 Components)')
plt.show()
```




```
# 6️⃣ Feature Selection
X = data.drop('math score', axis=1, errors='ignore')
y = data['math score']

X = X.select_dtypes(include=[np.number])
X = X - X.min()

selector = SelectKBest(score_func=chi2, k=5)
X_new = selector.fit_transform(X, y)
selected_features = X.columns[selector.get_support()]
print("\n✅ Selected Features:", list(selected_features))
```

```
✅ Selected Features: ['reading score', 'writing score']
/usr/local/lib/python3.12/dist-packages/sklearn/feature_selection/_univariate
  warnings.warn(
```

```
# ✅ Summary of Transformations
summary = pd.DataFrame({
    'Step': ['Data Exploration', 'Missing Data Handling', 'Encoding', 'S
    'Purpose': [
        'Understand structure',
        'Clean missing values',
        'Convert categorical to numeric',
        'Normalize numeric scale',
        'Reduce dimensionality',
        'Keep most important features'
```

```
    ]
})
print("\n📋 Summary of Transformations:")
print(summary)
```

```
📋 Summary of Transformations:
                  Step                         Purpose
0      Data Exploration           Understand structure
1  Missing Data Handling           Clean missing values
2               Encoding  Convert categorical to numeric
3                Scaling        Normalize numeric scale
4                    PCA          Reduce dimensionality
5      Feature Selection    Keep most important features
```