```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Step 1: Load the dataset
df = pd.read_csv("student_performance.csv")
```

```
# Step 2: Display first few rows
print("🔷 Dataset Preview:")
print(df.head())
```

```
🔷 Dataset Preview:
   Gender  Hours_Studied  Attendance  Sleep_Hours  Previous_Score  Final_Score
0    Male              5          80            6              70           75
1  Female              6          90            7              80           85
2    Male              2          60            5              50           52
3  Female              8          95            8              88           92
4    Male              4          70            6              60           65
```

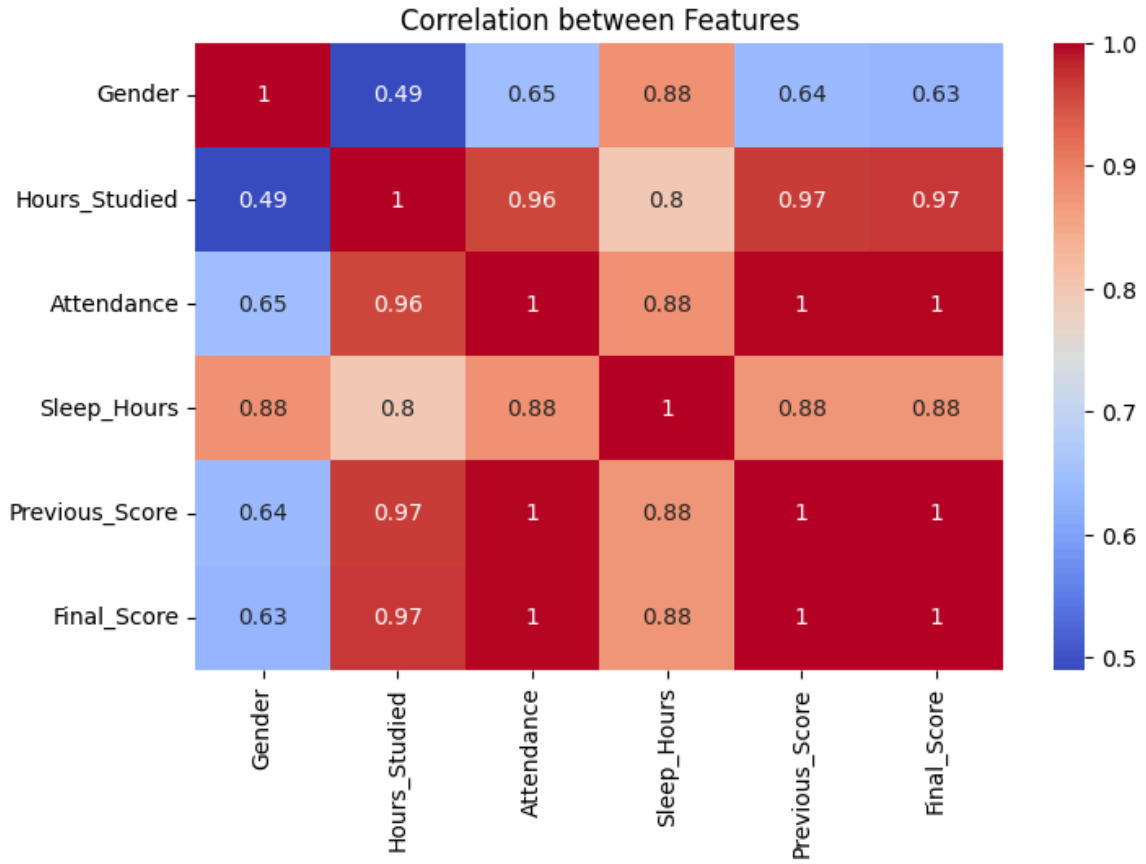
```
# Step 3: Basic Info
print("\n🔷 Dataset Info:")
print(df.info())
```

```
🔷 Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Gender          10 non-null     object
 1   Hours_Studied   10 non-null     int64
 2   Attendance      10 non-null     int64
 3   Sleep_Hours     10 non-null     int64
 4   Previous_Score  10 non-null     int64
 5   Final_Score     10 non-null     int64
dtypes: int64(5), object(1)
memory usage: 612.0+ bytes
None
```

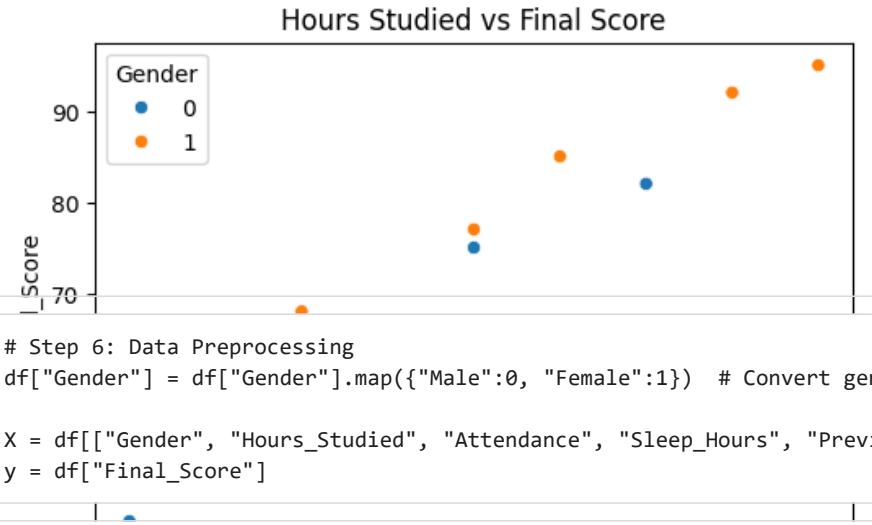
```
# Step 4: Check for missing values
print("\n🔷 Missing Values:")
print(df.isnull().sum())
```

```
🔷 Missing Values:
Gender            0
Hours_Studied     0
Attendance        0
Sleep_Hours       0
Previous_Score    0
Final_Score       0
dtype: int64
```

```
# Step 5: Data Visualization
plt.figure(figsize=(8,5))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation between Features")
plt.show()
```




```
plt.figure(figsize=(6,4))
sns.scatterplot(x="Hours_Studied", y="Final_Score", data=df, hue="Gender")
plt.title("Hours Studied vs Final Score")
plt.show()
```




```
# Step 6: Data Preprocessing
df["Gender"] = df["Gender"].map({"Male":0, "Female":1})  # Convert gender to numeric

X = df[["Gender", "Hours_Studied", "Attendance", "Sleep_Hours", "Previous_Score"]]
y = df["Final_Score"]
```

```
# Step 7: Split Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Step 8: Train Model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Step 9: Predict
y_pred = model.predict(X_test)
```

```
# Step 10: Evaluate Model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("\n🔷 Model Evaluation:")
print(f"Mean Squared Error: {mse:.2f}")
print(f"R² Score: {r2:.2f}")
```

```
🔷 Model Evaluation:
Mean Squared Error: 2.73
```

```
R² Score: 0.99
```

```
# Step 11: Compare Actual vs Predicted
comparison = pd.DataFrame({"Actual": y_test, "Predicted": np.round(y_pred, 2)})
print("\n🔷 Actual vs Predicted Final Scores:")
print(comparison)
```

```
🔷 Actual vs Predicted Final Scores:
   Actual  Predicted
8      45      42.82
1      85      84.17
```

```
# Step 12: Visualize Prediction
plt.figure(figsize=(6,4))
plt.scatter(y_test, y_pred, color='blue')
plt.xlabel("Actual Final Score")
plt.ylabel("Predicted Final Score")
plt.title("Actual vs Predicted Score")
plt.show()
```