[2]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

df = pd.read_csv("C:/MET/Level 4 Sem 1/Machine Learning/Task 3/calories.csv")

df.drop_duplicates(inplace=True)
df.dropna(inplace=True)

if "Gender" in df.columns:
    le = LabelEncoder()
    df["Gender"] = le.fit_transform(df["Gender"])
else:
    df["Gender"] = 0

X = df[["Age","Weight","Height","Duration","Heart_Rate","Body_Temp"]]
y = df["Calories"]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("MSE:", mean_squared_error(y_test, y_pred))
```

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("MSE:", mean_squared_error(y_test, y_pred))
print("R2 Score:", r2_score(y_test, y_pred))

new_data = pd.DataFrame([{
    "Age": 25,
    "Weight": 70,
    "Height": 175,
    "Duration": 60,
    "Heart_Rate": 130,
    "Body_Temp": 37
}])

new_scaled = scaler.transform(new_data)

prediction = model.predict(new_scaled)

print("Predicted Calories Burned:", prediction[0])
```

```
MSE: 132.0268480349755
R2 Score: 0.9672860085124584
Predicted Calories Burned: 493.93653832821917
```