

```

from sklearn.cluster import KMeans
import numpy as np

# Sample data with corrected labels (assuming 2 clusters)
X = np.array([[1.713, 1.586], [0.180, 1.786], [0.353, 1.240],
              [0.940, 1.566], [1.486, 0.759], [1.266, 1.106],
              [1.540, 0.419], [0.459, 1.799], [0.773, 0.186]])
y = np.array([0, 1, 1, 0, 0, 0, 1, 1, 1]) # Assuming 2 clusters (adjust if needed)

# Create the KMeans model with 3 clusters (or adjust as needed)
kmeans = KMeans(n_clusters=3, random_state=0).fit(X)

# Print the input data with labels (assuming labels are available)
print("The input data is:")
print("VAR1\tVAR2\tCLASS")
for i, val in enumerate(X):
    print(f"{val[0]}\t{val[1]}\t{y[i]}") # Use f-string for formatted printing

print("=" * 20)

# Get user input for test data
while True: # Loop until valid input is provided
    try:
        VAR1 = float(input("Enter Value for VAR1: "))
        VAR2 = float(input("Enter Value for VAR2: "))
        break
    except ValueError:
        print("Invalid input. Please enter numbers only.")

test_data = np.array([VAR1, VAR2])

# Predict the class for the test data
predicted_class = kmeans.predict([test_data])[0]
print(f"The predicted class is: {predicted_class}")

print("=" * 20)

```

```

➡ /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
The input data is:
VAR1    VAR2    CLASS
1.713    1.586    0
0.18     1.786    1
0.353    1.24     1
0.94     1.566    0
1.486    0.759    0
1.266    1.106    0
1.54     0.419    1
0.459    1.799    1
0.773    0.186    1
=====
Enter Value for VAR1: 0.906
Enter Value for VAR2: 0.606
The predicted class is: 0
=====

```