EXPERIMENT-5

CODE:-

```
import math
from collections import Counter

# ---------------- Dataset ----------------
# Format: [Feature1, Feature2, Class]
dataset = [
    [1, 2, 'A'],
    [2, 3, 'A'],
    [3, 3, 'A'],
    [6, 5, 'B'],
    [7, 7, 'B'],
    [8, 6, 'B']
]

# ---------------- Euclidean Distance ----------------
def euclidean_distance(x1, x2):
    return math.sqrt(sum((x1[i] - x2[i]) ** 2 for i in range(len(x1))))

# ---------------- K-NN Algorithm ----------------
def knn(dataset, test_sample, k):
    distances = []

    for data in dataset:
        distance = euclidean_distance(data[:-1], test_sample)
        distances.append((distance, data[-1]))
    distances.sort(key=lambda x: x[0])
    k_nearest = distances[:k]

    classes = [label for _, label in k_nearest]
    return Counter(classes).most_common(1)[0][0]

# ---------------- Test the Algorithm ----------------
test_sample = [5, 5]
k = 3

result = knn(dataset, test_sample, k)

print("Test Sample:", test_sample)
print("K value:", k)
print("Predicted Class:", result)
```

OUTPUT:-

```
Test Sample: [5, 5]
K value: 3
Predicted Class: B
```