

## EXPERIMENT 5:

```
import math

# Dataset: [features], class label
data = [
    ([1, 2], 'A'),
    ([2, 3], 'A'),
    ([3, 3], 'B'),
    ([6, 5], 'B')
]

k = 3
test = [2, 2]

# Euclidean distance
def distance(a, b):
    return math.sqrt(sum((x - y) ** 2 for x, y in zip(a, b)))

# Compute distances
distances = []
for x, label in data:
    distances.append((distance(test, x), label))

# Sort by distance
distances.sort()

# Get k nearest labels
neighbors = [label for _, label in distances[:k]]
```

```
# Majority voting
result = max(set(neighbors), key=neighbors.count)

print("Test Point:", test)
print("Predicted Class:", result)
```

### Output

```
Test Point: [2, 2]
Classified as: A
```

```
=== Code Execution Successful ===
```