Experiment -16

```
import math

# Simple dataset: [feature1, feature2, class]
data = [
    [1, 2, 0],
    [2, 3, 0],
    [3, 3, 1],
    [6, 5, 1]
]

test = [2, 2]
actual = 0

# ---------- Naive Bayes ----------
def naive_bayes():
    class0 = [d for d in data if d[-1] == 0]
    class1 = [d for d in data if d[-1] == 1]
    return 0 if len(class0) > len(class1) else 1

# ---------- KNN ----------
def knn(k=3):
    dist = []
    for d in data:
        dist.append((math.dist(test, d[:-1]), d[-1]))
    dist.sort()
    labels = [l for _, l in dist[:k]]
    return max(set(labels), key=labels.count)

# ---------- Decision Tree (Rule-based) ----------
def decision_tree():
    return 0 if test[0] < 3 else 1

# Predictions
nb = naive_bayes()
kn = knn()
dt = decision_tree()

print("Naive Bayes Prediction:", nb)
print("KNN Prediction:", kn)
print("Decision Tree Prediction:", dt)

# Accuracy
```

```
print("\nAccuracy:")
print("Naive Bayes:", int(nb == actual))
print("KNN:", int(kn == actual))
print("Decision Tree:", int(dt == actual))
```

Output: