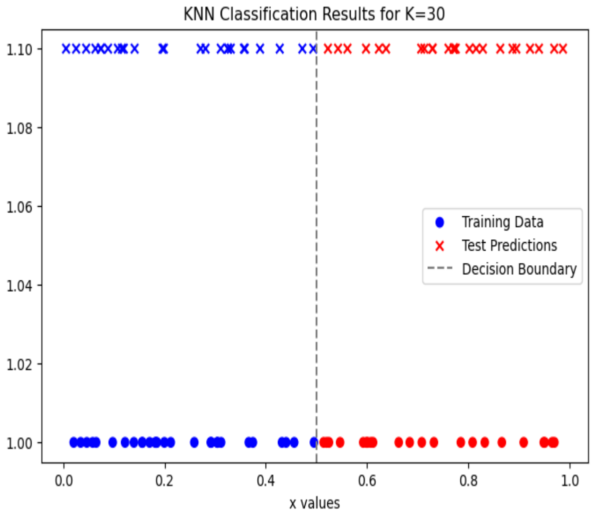
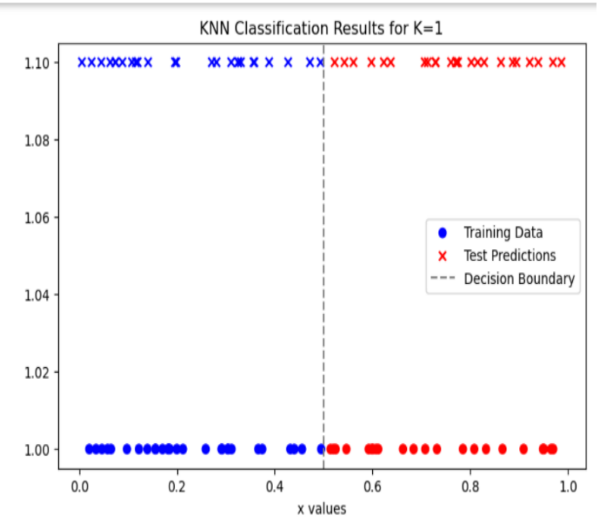
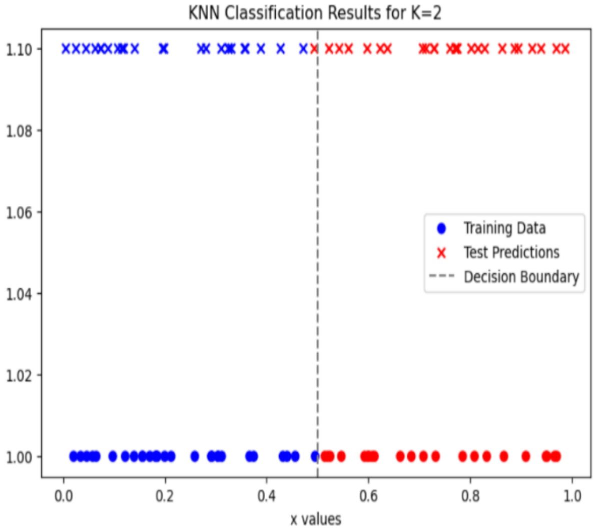
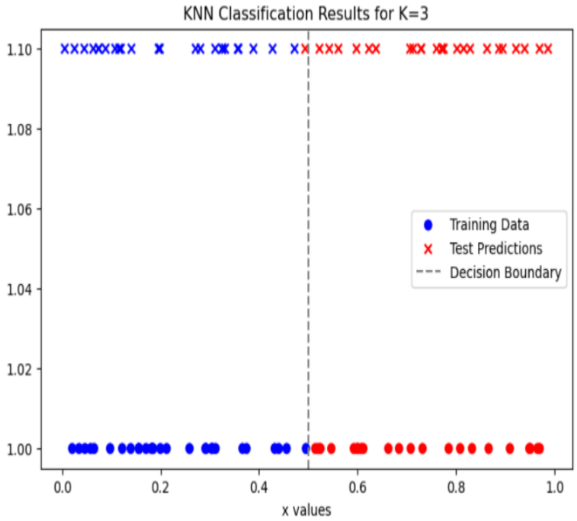
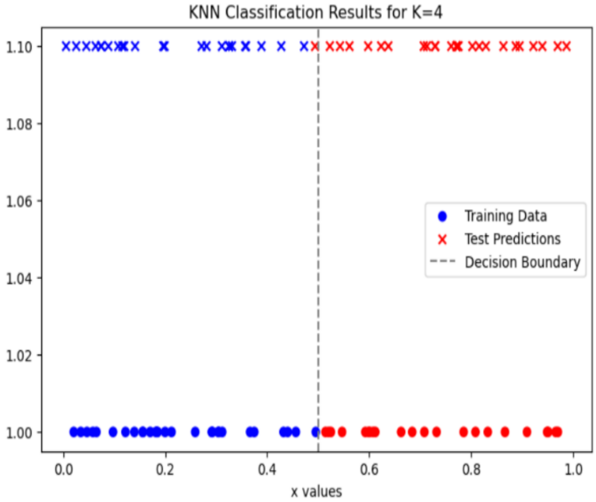
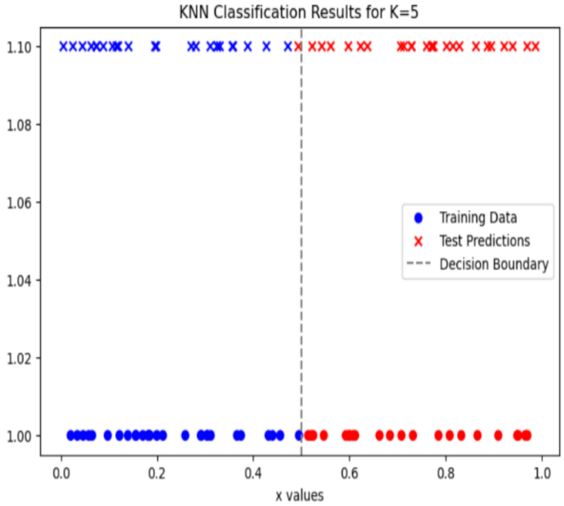

KNN Classification Results for K=30
1.10
1.08
1.06
1.04
1.02
1.00
0.0
0.2
0.4
0.6
0.8
1.0
x values
● Training Data
× Test Predictions
--- Decision Boundary

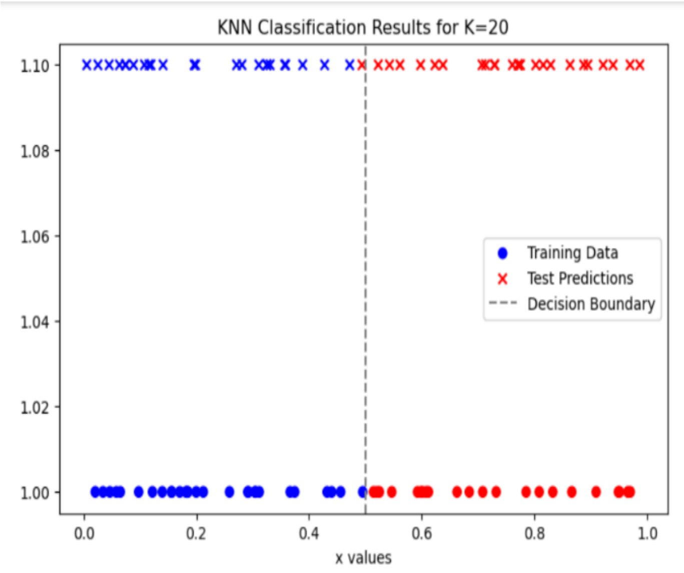

KNN Classification Results for K=1
1.10
1.08
1.06
1.04
1.02
1.00
0.0
0.2
0.4
0.6
0.8
1.0
x values
● Training Data
× Test Predictions
--- Decision Boundary

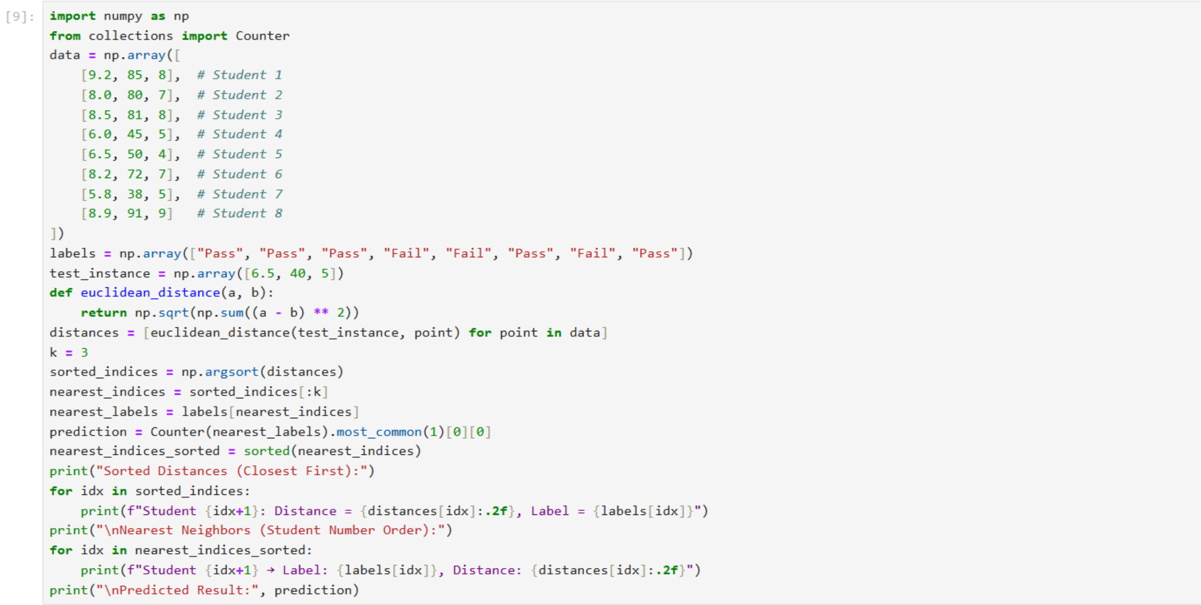

KNN Classification Results for K=2
1.10
1.08
1.06
1.04
1.02
1.00
0.0
0.2
0.4
0.6
0.8
1.0
x values
● Training Data
× Test Predictions
--- Decision Boundary


KNN Classification Results for K=3
1.10
1.08
1.06
1.04
1.02
1.00
0.0
0.2
0.4
0.6
0.8
1.0
x values
● Training Data
× Test Predictions
--- Decision Boundary

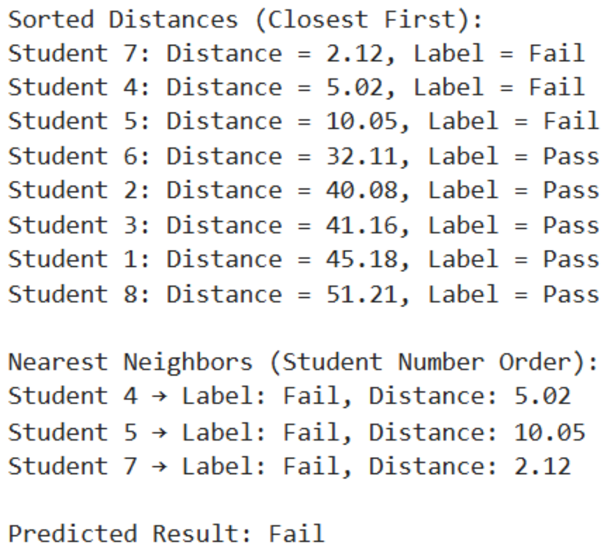

KNN Classification Results for K=4
1.10
1.08
1.06
1.04
1.02
1.00
0.0
0.2
0.4
0.6
0.8
1.0
x values
● Training Data
× Test Predictions
--- Decision Boundary


KNN Classification Results for K=5
1.10
1.08
1.06
1.04
1.02
1.00
0.0
0.2
0.4
0.6
0.8
1.0
x values
● Training Data
× Test Predictions
--- Decision Boundary





```
[9]: import numpy as np
     from collections import Counter
     data = np.array([
         [9.2, 85, 8],  # Student 1
         [8.0, 80, 7],  # Student 2
         [8.5, 81, 8],  # Student 3
         [6.0, 45, 5],  # Student 4
         [6.5, 50, 4],  # Student 5
         [8.2, 72, 7],  # Student 6
         [5.8, 38, 5],  # Student 7
         [8.9, 91, 9]   # Student 8
     ])
     labels = np.array(["Pass", "Pass", "Pass", "Fail", "Fail", "Pass", "Fail", "Pass"])
     test_instance = np.array([6.5, 40, 5])
     def euclidean_distance(a, b):
         return np.sqrt(np.sum((a - b) ** 2))
     distances = [euclidean_distance(test_instance, point) for point in data]
     k = 3
     sorted_indices = np.argsort(distances)
     nearest_indices = sorted_indices[:k]
     nearest_labels = labels[nearest_indices]
     prediction = Counter(nearest_labels).most_common(1)[0][0]
     nearest_indices_sorted = sorted(nearest_indices)
     print("Sorted Distances (Closest First):")
     for idx in sorted_indices:
         print(f"Student {idx+1}: Distance = {distances[idx]:.2f}, Label = {labels[idx]}")
     print("\nNearest Neighbors (Student Number Order):")
     for idx in nearest_indices_sorted:
         print(f"Student {idx+1} → Label: {labels[idx]}, Distance: {distances[idx]:.2f}")
     print("\nPredicted Result:", prediction)
```



```
Sorted Distances (Closest First):
Student 7: Distance = 2.12, Label = Fail
Student 4: Distance = 5.02, Label = Fail
Student 5: Distance = 10.05, Label = Fail
Student 6: Distance = 32.11, Label = Pass
Student 2: Distance = 40.08, Label = Pass
Student 3: Distance = 41.16, Label = Pass
Student 1: Distance = 45.18, Label = Pass
Student 8: Distance = 51.21, Label = Pass

Nearest Neighbors (Student Number Order):
Student 4 → Label: Fail, Distance: 5.02
Student 5 → Label: Fail, Distance: 10.05
Student 7 → Label: Fail, Distance: 2.12

Predicted Result: Fail
```