**Submission sheet**

Assignment IV

**TASK 0: Warm-up**

Number of instances: 1200

Number of attributes: 2

Number of instances in each class: A 240 N 960

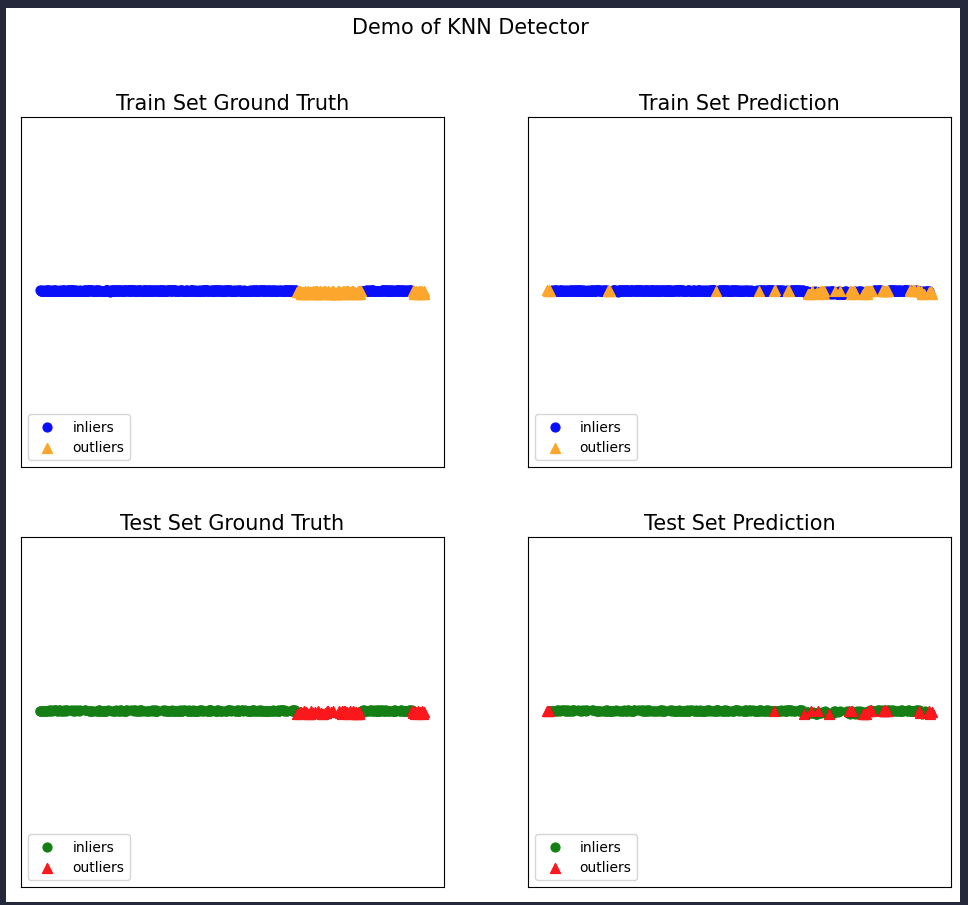
**TASK 1: KNN-Based Anomaly Detection**

Performance results:

1) AUC-ROC: 0.82

2) Average Precision 0.47

Visualizations results (Please paste the images you have generated in Task1):



# TASK 2: Parameter sensitivity

Which “n\_neighbors” and “method” correspond to the best performance? Write the number of neighbors and the performance score for the specified metric:

1) n\_neighbors and method: 20 largest best AUC-ROC: 0.9695

1) n\_neighbors and method: 20 largest best Average Precision: 0.8691

# TASK 3: Local Outlier Factor

Which n\_neighbors correspond to the best performance? Write the number of neighbors and the performance score for the specified metric:

1) n\_neighbors: 20 best AUC-ROC: 0.6

1) n\_neighbors: 20 best Average Precision: 0.31

According to the results from trying different n\_neighbours, which algorithm (KNN or LOF) is more sensitive to hyperparameters? Please explain your findings below.

LOF is more sensitive to hyperparameters than KNN. LOF's performance metrics fluctuated significantly with different n\_neighbors values, indicating high sensitivity and a need for precise tuning. While KNN's performance remained more stable across various n\_neighbors due to its inherent flexibility.

# TASK 4: Real-world Anomaly Detection

Please briefly explain which algorithms and metrics you have chosen for this task. Report and explain the results you have gathered from your analysis. Have you noticed any difference between the metrics used and the chosen algorithm? Which algorithm seems more appropriate for this task?

We chose LOF, Isolation Forest and OCSVM because they’re well-known anomaly detection methods that was also covered in class. LOF was picked because it detects anomalies by looking at local density differences. Isolation Forest works well when randomly isolating anomalies, and OCSVM is good at drawing boundaries around normal data to spot anything that stands out.

We used ROC-AUC and Average Precision to evaluate them. Results:

LOF: ROC-AUC 0.56, Average Precision 0.17

Isolation Forest: ROC-AUC 0.93, Average Precision 0.58

OCSVM: ROC-AUC 0.95, Average Precision 0.57

LOF didn’t do well, so it’s probably not a great fit for this dataset. Isolation Forest and OCSVM performed much better, with OCSVM having the highest ROC-AUC. The ROC-AUC scores were higher than Average Precision for all of them, which is normal for imbalanced data. Overall, OCSVM seems like the best choice since it did the best at separating anomalies from normal data.

**TASK 5 (OPTIONAL): Improving Detection Performance**

Please briefly explain which strategies you have tried for improving performance. Did they work? If not, can you explain why?