# Exercise 7 – Evaluation III

**Introduction to Machine Learning**

*Hint: Useful libraries*

### R

```
# Consider the following libraries for this exercise sheet:

library(ggplot2)
```

### Python

```
# Consider the following libraries for this exercise sheet:

import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics
```

## Exercise 1: ROC metrics

> Learning goals
>
> 1. Create confusion matrices and compute associated evaluation metrics
> 2. Compute ROC coordinates and AUC
> 3. Understand relationship between ROC curve & classification threshold

Consider a binary classification algorithm that yielded the following results:

| ID | True class | Prediction |
|---|---|---|
| 1 | 0 | 0.33 |
| 2 | 0 | 0.27 |
| 3 | 0 | 0.11 |
| 4 | 1 | 0.38 |
| 5 | 1 | 0.17 |
| 6 | 1 | 0.63 |
| 7 | 1 | 0.62 |
| 8 | 1 | 0.33 |
| 9 | 0 | 0.15 |
| 10 | 0 | 0.57 |

---

Create a confusion matrix assuming a threshold of 0.5. Point out which values correspond to true positives (`TP`), true negatives (`TN`), false positives (`FP`), and false negatives (`FN`).

---

Calculate: PPV, NPV, TPR, FPR, ACC, MCE and $F1$ measure.

---

Draw the ROC curve and interpret it.

---

Calculate the AUC.

---

How would the ROC curve change if you had chosen a different threshold in a)?

## Exercise 2: $k$-NN

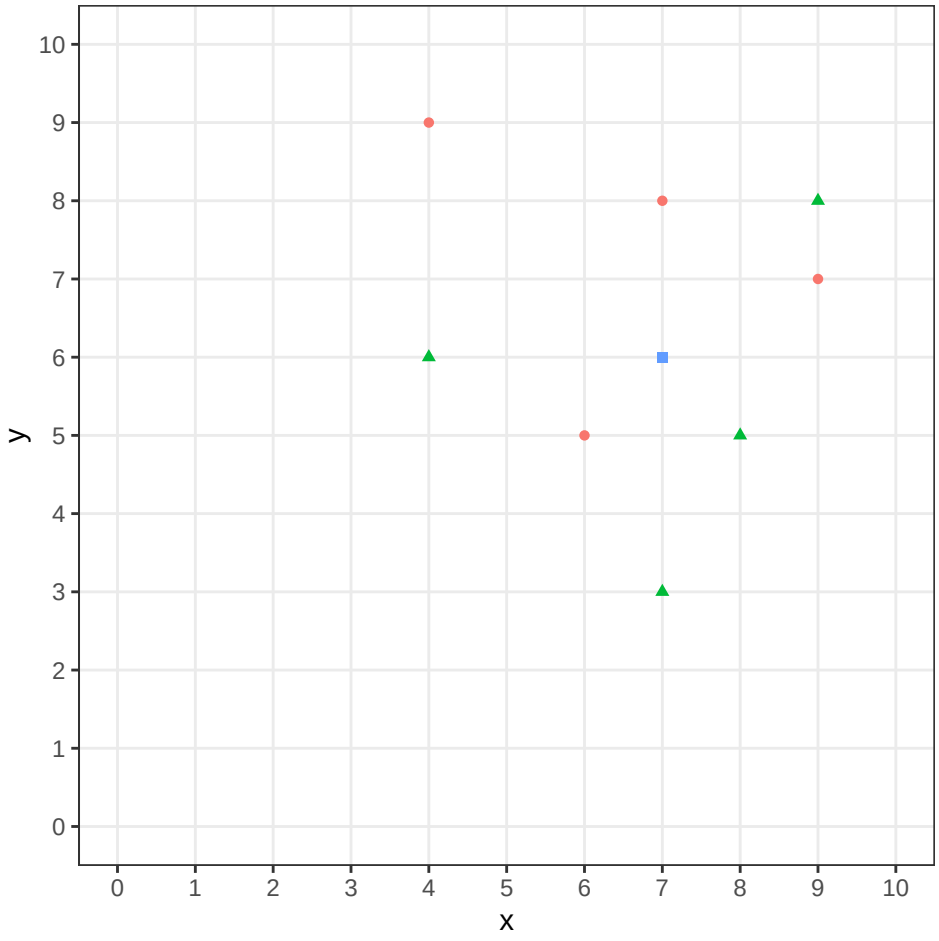
Learning goals

1. Perform k-NN by visual means
2. Perform k-NN with pen and paper, possibly using weighted distances

Let the two-dimensional feature vectors in the following figure be instances of two different classes (triangles and circles). Classify the point (7, 6) – represented by a square in the picture – with a $k$-NN classifier using $L1$ norm (Manhattan distance):

$$d_{\text{Manhattan}}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{j=1}^{p} |x_j - \tilde{x}_j|.$$

As a decision rule, use the unweighted number of the individual classes in the $k$-neighborhood, i.e., assign the point to the class that represents most neighbors.



i. $k = 3$

ii. $k = 5$

iii. $k = 7$

Now consider the same constellation but assume a regression problem this time, where the circle-shaped points have a target value of 2 and the triangles have a value of 4.

Again, predict for the square point (7, 9), using both the *unweighted* and the *weighted* mean in the neighborhood (still with Manhattan distance).

*Hint*

We now consider both *unweighted* and *weighted* predictions. Recall that weights are computed based on the distance between the point of interest and its respective neighbors. With the Manhattan, or "city block" metric, the distance can be read from the plot by walking along the grid lines (shortest way). For example, in the 3-neighborhood, all points have a distance of 2 from our square, so all get weights $\frac{1}{2}$.

i. $k = 3$

ii. $k = 5$

iii. $k = 7$