# Data Science
# Exercise 5 - Comparative Experimentation

Student: se21m024, Thomas Stummer

The source code can be found in the document ***se21m024_Stummer_ex5_comp_exp.ipynb***.

Small data set: Heart Failure Prediction
The data set was provided by Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020) (https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-1023-5) and downloaded from https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data.

Big data set: Covertype
The data set was provided by Jock A. Blackard and Colorado State University and downloaded from https://archive.ics.uci.edu/ml/datasets/Covertype.

# Small data set: Heart Failure Prediction

The data was split into input features a target feature. The target feature is 'DEATH_EVENT' that indicates either the person has died. The column 'time' was not used as input feature due to the direct connection to the target feature 'death_event' according to https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data/discussion/178372. The train/test split was chosen to be 2/3 to 1/3 as required. For the k-NN apporach, k-d tree was chosen as algorithm to gain results within a reasonable amount of time.

## Results table

| **Algorithm with parameters** | **Accuracy** | **F1** | **Training time** | **Testing time** |
|---|---|---|---|---|
| k-NN (5-NN) | 0.616 | 0.549 | 0.001997 sec | 0.003997 sec |
| k-NN (10-NN) | 0.677 | 0.554 | 0.001999 sec | 0.004003 sec |
| k-NN (15-NN) | 0.687 | 0.577 | 0.002001 sec | 0.003998 sec |
| Perceptron (alpha: 0.0001) | 0.687 | 0.559 | 0.002 sec | 0.000998 sec |
| Perceptron (alpha: 0.001) | 0.687 | 0.559 | 0.002 sec | 0.001001 sec |
| Perceptron (alpha: 0.01) | 0.687 | 0.559 | 0.002 sec | 0.000999 sec |
| Decision tree (max features: None) | 0.697 | 0.702 | 0.003001 sec | 0.001 sec |
| Decision tree (max features: sqrt) | 0.626 | 0.628 | 0.001999 sec | 0.000998 sec |
| Decision tree (max features: log2) | 0.626 | 0.628 | 0.001 sec | 0.001 sec |

## Interpretation

The highest (= best) accuracy and F1 measures were accomplished by the decision tree with unlimited number of features for consideration for each split. Other max feature settings resulted in lower accuracy and F1 measure.
The best accuracy reached with the perceptron is equal to the best accuracy reached with the k-NN algorithm with k=15. Changing the alpha value of the perceptron had no significantly impact on the accuracy or F1 measure. The highest F1 measure accomplished by the k-NN is a little higher than the one accomplished by the perceptron.

The training time was very similar for all algorithms and all parameter settings with about 0.002 seconds. Only the decision tree with no maximum feature amount took significantly longer (0.003 seconds) and the decision tree with maximum features 'log2' took significantly shorter (0.001 seconds).
The testing time for all k-NN runs was about 0.004 seconds which is significantly higher than the testing time for the perceptron and the decision tree which was about 0.001 seconds.

# Big data set: Covertype

The data was split into input features and a target feature. The target feature is 'Forest cover type class' than can be any value between 1 and 7 and indicates which type of vegetation is growing there mainly. The train/test split was chosen to be 2/3 to 1/3 as required. For the k-NN approach, k-d tree was chosen as algorithm to gain results within a reasonable amount of time.

## Results table

| **Algorithm with parameters** | **Accuracy** | **F1** | **Training time** | **Testing time** |
|---|---|---|---|---|
| k-NN (5-NN) | 0.965 | 0.965 | 12.033436 sec | 20.152396 sec |
| k-NN (10-NN) | 0.955 | 0.954 | 12.148761 sec | 24.076917 sec |
| k-NN (15-NN) | 0.946 | 0.946 | 12.482787 sec | 31.48592 sec |
| Perceptron (alpha: 0.0001) | 0.584 | 0.563 | 9.840077 sec | 0.047 sec |
| Perceptron (alpha: 0.001) | 0.584 | 0.563 | 9.665985 sec | 0.041996 sec |
| Perceptron (alpha: 0.01) | 0.584 | 0.563 | 9.708394 sec | 0.045001 sec |
| Decision tree (max features: None) | 0.933 | 0.933 | 5.58124 sec | 0.096001 sec |
| Decision tree (max features: sqrt) | 0.877 | 0.877 | 1.212917 sec | 0.084999 sec |
| Decision tree (max features: log2) | 0.87 | 0.87 | 0.959233 sec | 0.095001 sec |

## Interpretation

The highest (= best) accuracy and F1 measures were accomplished by the k-NN algorithm with k=5. Higher k resulted in slightly worse results.
The decision tree delivered the seconds best results (unlimited feature number performed best) regarding accuracy and F1 measure. The perceptron reached far lower accuracy and F1 measure (independent of the alpha value used).

The decision tree was by far the fastest to train (especially with max features set to 'log2'). With a great gap, perceptron and k-NN followed.
Regarding the testing time the perceptron was fastest (half the time was required compared to the decision tree). The testing time of the k-NN was many magnitudes higher than for the two other algorithms.

# Comparison between data sets

For the small data set the decision tree delivered the most accurate results, for the big data set the k-NN algorithm performed best.
While the time required for training and testing did not vary much among the algorithms on the small data set, a huge difference could be spotted on the big data set when comparing the efficiency for the different algorithms.