Contents

[01 Machine Learning Fundamentals 2](#_Toc183332898)

[02 Regression 3](#_Toc183332899)

[03 Tree-based Models 4](#_Toc183332900)

[04 KNN and Logistic Regression 5](#_Toc183332901)

[05 Naïve Bayes and Support Vector Machines 6](#_Toc183332902)

[06 Data Preparation and Feature Engineering 7](#_Toc183332903)

[07 Validation Methods and Performance Metrics 8](#_Toc183332904)

[08 Dimensionality reduction 9](#_Toc183332905)

[09 Clustering 10](#_Toc183332906)

[10 Neural Networks 1 11](#_Toc183332907)

[11 Neural Networks 2 12](#_Toc183332908)

# 01 Machine Learning Fundamentals

It is very important that you attend this session as all the practical and formal information will be given. The lecture will be held by R. Brooks. Expect it to take up all 4 lessons.

In this session we introduce the course, the assignments, and will go through how to complete a Machine Learning project from defining the problem to building the final model:

* Explain what is meant by the term Machine Learning (ML)
* Explain what is meant by supervised vs. unsupervised learnin
* Explain the overall difference between classification and regressio
* Describe the "train-test" methodology
* Introduction to the importance of mathematics in machine learning
* The role of linear algebra in representing data and models.
* Intro to Numpy, Pandas and DataFrames and working with Matrices as objects.

After attending this lecture, reading the corresponding part of the book and doing exercises, I expect you to be able to:

* Define and articulate the scope of a Machine Learning problem, including understanding the big picture and setting clear objectives
* Understand why it is essential to perform model optimization and fine-tuning techniques to improve performance and achieve robustness in predictions, including understanding the importance of hyperparameter tuning.
* Understand why mathematics, particularly linear algebra, calculus, and statistics, is fundamental to machine learning and its optimization processes
* Gain knowledge on how linear algebra facilitates the representation of machine learning data, models, and computations using arrays and matrices.

# 02 Regression

* Linear regression algorithms:
  + Ordinary Least Squares (OLS) regression
  + Ridge regression
  + Lasso regression
* R² performance metric for regression.
* Gradient Descent.

**Learning Outcomes**

* Explain what is meant by "regression" and in which contexts to apply it
* Explain the following linear regression models, their strengths and weaknesses, and apply them in python:
  + Ordinary Least Squares (OLS) regression
  + Ridge regression
  + Lasso regression
  + Elastic Net Regression
* Explain what is meant by the term "regularization" in an ML-context
* Describe what is meant by "bias" and "variance" in relation to ML-algorithms
* Explain the R²-metric for evaluating the performance of a linear regression algorithm
* Explain the MSE metric
* Explain how regression models can be trained

# 03 Tree-based Models

**Topics**

* Decision trees
* Random forests
* Gradient-boosted decision trees

**Learning Outcomes**

* Use and implement decision trees, random forests and gradient boosted decision trees in python.
* Describe the advantages and disadvantages of using decision trees, random forests and gradient boosted decision trees, respectively.
* Visualize decision trees in different ways.
* Extract and interpret feature importance.
* Describe how the Gini impurity index can be used to determine which feature to branch off on.
* Explain what is meant by pre-pruning.
* Explain how random forests are random, including what is meant by bootstrapping and feature selection in this context.
* Explain what is meant by soft voting.
* Discuss different hyperparameters of tree-based methods, and how tuning these parameters influence the results.

# 04 KNN and Logistic Regression

**Topics**

* Accuracy
* Logistic Regression
* k-Nearest Neighbour

**Learning Outcomes**

* Train a logistic regression (LR) and k Nearest Neighbours (kNN) algorithm on a dataset in sklearn
* Explain the principles behind the kNN algorithm
* Explain the key ideas behind logistic regression and implement a logistic regression classifier in python.
* Explain the concept of “maximum likelihood” and how it is used.
* Explain and use L1 and L2 regularization in the context of logistic regression, and discuss the difference between these approaches, as well as the importance of the hyperparameter C.
* Discuss advantages and disadvantages of logistic regression.
* Explain what is meant by the "hyperparameters" of an algorithm

# 05 Naïve Bayes and Support Vector Machines

**Topics**

This lecture will delve into more sophisticated classification methods within machine learning. We will explore the theory and application of Support Vector Machines (SVM) and Naive Bayes classifiers.

**Learning Outcomes**

* Explain support vectors for linearly separable data, and how support vectors influence the decision boundary.
* Explain and exemplify how adding new features can make non-linearly separable data linearly separable.
* Discuss the key ideas behind the kernel trick and how this is used in kernelized support vector machines.
* Discuss how, when using a Gaussian kernel, the hyperparameters C and γ influence the decision boundaries.
* Discuss advantages and disadvantages of support vector machines.
* Understand the probabilistic foundations of the Naive Bayes classifier and its assumptions about feature independence.
* Describe the different types of Naive Bayes classifiers (e.g., Gaussian, Multinomial) and their appropriate application contexts.
* Apply a Naive Bayes classifier to text data and other types of datasets using sklearn.
* Discuss the concept of hyperparameters in the context of SVM and Naive Bayes, and demonstrate how to tune them to improve model performance.

# 06 Data Preparation and Feature Engineering

**Topics**

* Data cleaning: What to do with missing values, outliers, nonsensical values
* Box Plots and other exploratory visualisations
* Data imputation
* Feature engineering:
  + Dummy variables
  + Scaling

**Learning Outcomes**

* Know how to approach features and when to drop and/or engineer them for your specific purposes
* Prepare a dataset for ML. Specifically, you should be able to explain and perform each of the operations below:
  + Handle missing values/NaN-values in appropriate ways
  + Identify and handle outliers using a boxplot
  + Create dummy variables
  + Scale/normalize variables
  + Create a bag-of-words-representation for text-data

# 07 Validation Methods and Performance Metrics

**Topics**

* The train/test-methodelogy
* The validation set methodology
* The cross-validation methodology
* The leave-one-out-methodology
* Accuracy
* Confusion matrix
* Recall
* Precision
* F1-score
* Precision-recall-curve

**Learning Outcomes**

* Describe the "validation set"-methodology
* Describe the "cross validation"-methodology
* Describe the "leave one out"-methodology
* Apply each of the 3 methodologies above in sklearn
* Do hyperparameter tuning in sklearn using each of the 3 methodologies above (e.g. using the GridSearchCV-function in sklearn)
* Explain and calculate (in python) the following performance metrics for supervised classification:
  + Confusion matrix
  + Accuracy
  + Recall ( = "True Positive Rate" (TPR) )
  + Precision ( = "Positive Prediction Rate" (PPR) )
  + F1-score
  + Precision-recall-curve

# 08 Dimensionality reduction

**Topics**

* Principal component analysis (PCA)
* t-distributed stochastic neighbor embedding (t-SNE)

**Learning Outcomes**

* Use principle component analysis (PCA) to reduce the dimensions of your dataset
* Describe how PCA can be used for clustering analyses
* Create 2-dimensional clustering-plots in python using PCA and t-SNE

# 09 Clustering

**Topics**

* k means
* Hierarchical clustering
* DBSCAN

**Learning Outcomes**

* Describe the following clustering algorithms along with their advantages and disadvantages:
  + k-Means
  + Agglomerative clustering
  + DBSCAN
* Apply the above clustering algorithms in Python
* Evaluate clustering algorithms (e.g. by inspecting the output in 2D-plots or in other ways inspecting which elements are clustered together).
* Use a dendrogram to determine the optimal number of clusters

# 10 Neural Networks 1

**Topics**

Fundamental aspects of neural networks.

**Learning Outcomes**

* Explain what is meant by artificial neurons, and how these are linked to form artificial neural networks.
* Explain what a perceptron is, and how it transforms an input vector to an output, including the importance of weights and biases.
* Discuss how to structure a neural network.
* Discuss and summarize the method of (stochastic) gradient descent and how it is used to train a neural network, including how the learning rate influences results.
* Reflect upon the problems caused by using (a) perceptrons as artificial neurons and (b) the accuracy as the metric to optimize during model training, including how these problems can be solved using activation and loss functions, respectively.
* Sketch different activation functions, including the sigmoid, tanh and ReLU functions, and discuss why the softmax activation function is generally used in the output layer.
* Implement a neural network in python using the tensorflow.keras module.

# 11 Neural Networks 2

**Topics**

* Convolutional Neural Networks
* Pre-trained networks

**Learning Outcomes**

* Explain what is meant by artificial neurons, and how these are linked to form artificial neural networks.
* Explain what a perceptron is, and how it transforms an input vector to an output, including the importance of weights and biases.
* Discuss how to structure a neural network.
* Discuss and summarize the method of (stochastic) gradient descent and how it is used to train a neural network, including how the learning rate influences results.
* Reflect upon the problems caused by using (a) perceptrons as artificial neurons and (b) the accuracy as the metric to optimize during model training, including how these problems can be solved using activation and loss functions, respectively.
* Sketch different activation functions, including the sigmoid, tanh and ReLU functions, and discuss why the softmax activation function is generally used in the output layer.
* Implement a neural network in python using the tensorflow.keras module.