Data Science and Machine Learning in Python

Stephan Weyers

Fachhochschule Dortmund
University of Applied Sciences and Arts

Topics covered in the online lectures

Fachhochschule Dortmund
University of Applied Sciences and Arts

Part 1: Data Science

	Date	Topics covered
1	Apr 13th	Course introduction Data Science motivation How to use Jupyter Notebook Python types and lists Loops, if/else, functions
2	Apr 20th	Python tuples, lists, dictionaries Functions Numpy basics, operations Image processing
3	Apr 27th	Pandas Series, DataFrame Pandas basic operations Import/export files
4	May 4th	Principles of data visualization Data cleaning and preparation Join, combine and reshape data
5	May 11th	Volkswohl Bund dataset Data visualization in Python How to write Data Science reports Data aggregation and grouping

Part 2: Machine Learning

	Date	Topics covered
6	Jun 1st	Introduction to supervised learning Classification and regression scikit-learn k-Nearest Neighbors Linear regression (ridge and lasso)
7	Jun 8th	Linear classification models Decision trees Random forests and gradient boosting
8	Jun 15th	Kernel support vector machines Neural networks Introduction to unsupervised learning Preprocessing and scaling Dimensionality reduction Principal component analysis
9	Jun 22nd	k-means clustering Hierarchical clustering DBSCAN
10	Jun 29th	Representing data Engineering features
11	Jul 6th	Model evaluation and improvement Text data analysis

Deadlines for Submission and Distribution of Grading

Student task	Deliverables	Deadline	Work	Share of grade
W01 Assignment	Code and results	Apr 26th	Team A	5.0%
W02 Case Study	Code / presentation slides	May 22nd	Team B	18.0%
W02 Case Study	Peer review*	May 31st	Individual	2.0%
W03 Assignment	Code and results	May 29th	Team B	5.0%
W04 Assignment	Code and results	Jun 12th	Team C	10.0%
W05 Assignment	Code and results	Jun 26th	Team D	7.0%
W06 Assignment	Code and results	Jul 3rd	Team D	13.0%
W07 Case Study	Code / presentation slides	Jul 17th	Team D	22.0%
W07 Case Study	Peer review*	Jul 31st	Individual	3.0%
DataCamp 1	Finish course	May 9th	Individual	2.5%
DataCamp 2	Finish course	May 30th	Individual	2.5%
DataCamp 3	Finish course	Jun 20th	Individual	2.5%
DataCamp 4	Finish course	Jul 11th	Individual	2.5%

* Peer review is mandatory. Quality of peer review itself is graded. Not providing peer review at all would result in high point deduction

Teams for assignment W04

Team	Univ.	Name
C1	FHDO	Daniel Tobien
C1	UBA	Lucía Ailén Kasman
C1	UBA	Francisco Alan Luna
C1	UV	Dietrich Ganz
C2	FHDO	Arnold Urbanio Olympio
C2	UDEM	Valentina Torres Torres Luján
C2	UGTO	Julio Campos Pérez
C2	UV	Jose Ignacio Meneses Castillo
C3	ESAN	Luiggy Johan Zea Guzman
C3	FHDO	Fabian Herberholt
C3	UV	Paula Toro
C3	UV	Sofia Contreras Figueroa
C4	FHDO	Mamadama Cherif
C4	UBA	Andrómeda P. Ovalles Castro
C4	UTTEC	Cesar Bravo Robles
C4	UV	Fernando Parada
C5	FHDO	Robin Drabon
C5	UBA	Mateo Agustín Fernández
C5	UGTO	Andrea Ortiz Alvarado
C5	UV	Franco Garrido
C6	FHDO	Jannick Bröring
C6	UBA	Juan Cruz Camacho
C6	UDEM	Thomas Jaramillo Vanegas
C6	UV	Jaime Godoy
C7	ESAN	Juan Jose A. Velasquez Leon
C7	UDEM	Maria José Morales Aranda
C7	UV	Valentina Andrea Acuña Ponce
C7	UV	Adonis Nicola Cruz Navarrete
C8	FHDO	Celine Cramer
C8	UBA	Daniel Kundro
C8	UBA	Belen Ticona
C8	UV	Nilari Berger Díaz

Team	Univ.	Name
C9	ESAN	María Ximena Latorre Guzmán
C9	FHDO	Marius Meiners
C9	UBA	Matías Nicolás Pereyra
C9	UBA	Manuel Durán
C10	ESAN	Alexis F. Huaman Fernandez
C10	FHDO	Intissar Boudi
C10	UBA	Lucas Trabanco
C10	UGTO	Frida Martinez Flores
C11	FHDO	Jessica Heilig
C11	UDEM	Jordana L.M. Apolinario Simon
C11	UV	Luis Martinez
C11	UV	Paula Riquelme
C12	FHDO	Marco Vom Bovert
C12	FHDO	Bedirhan Abaz
C12	UBA	Manuel Cabeza Galucci
C12	UV	Alejandra Valencia
C13	FHDO	Jakub Bogusz
C13	UBA	Francisco Rossi
C13	UBA	Victoria Cambriglia
C13	UV	Diego Del Rio
C13	UV	Rodrigo Llano Orellana
C14	FHDO	René Frackmann
C14	UBA	Sofía Nieva
C14	UTTEC	José Luís Godínez Vázquez
C14	UV	Maximiliano Arancibia Santana
C14	UV	Benjamin Serra
C15	FHDO	Mohamed Elbaraka
C15	UBA	Facundo Ignacio Zanalda
C15	UDEM	Dilan Stiven Correa López
C15	UV	Joel Santana
C15	UV	Lilian Torres

Team	Univ.	Name
C16	ESAN	Nayely Mayli Ore Ichpas
C16	ESAN	Jhossy J. Vargas Saldaña
C16	FHDO	Minh Quan Dinh
C16	UBA	Rocío Palacín Roitbarg
C16	UV	Manuel Orellana Hinojosa
C17	FHDO	Tegar Fathir Muhammad
C17	UBA	Gian Franco Lancioni
C17	UBA	Kevin Michalewicz
C17	UV	Felipe Galdames
C17	UV	Jorge Rodriguez
C18	FHDO	Justin Skupsch
C18	FHDO	Marco Kusnierek
C18	UBA	Victoria Marquez
C18	UDEM	Mariana Gómez Gómez
C18	UV	Paula Piña
C19	UDEM	Pablo Esteban Alzate Giraldo
C19	UGTO	Andrea Rodriguez Sotelo
C19	UTTEC	Hugo Isaac Vázquez Gutiérrez
C19	UV	Catalina Escobar
C19	UV	Dian Arriagada
C20	ESAN	Angela Karin Paredes Solano
C20	UBA	Joaquin Ceppi
C20	UGTO	Abraham Morales Iturriaga
C20	UV	Marcelo Leiton
C20	UV	Amaya Arroyo

Agenda for online lecture 7

Session	Topic	Mode	Materials used	Minutes	End
14:30-16:00	Organizational questions	Q&A		10	14:40
	Happiness data	Team work in break-out rooms	Lecture 06d notebook	45	15:25
	k-Nearest Neighbors	Lecture / Q&A	Lecture slides	5	15:30
	Linear classification	Lecture / Q&A	Lecture slides	25	15:55
16:10-17:40	MNIST show example	Lecture / Q&A	Lecture 07a notebook	20	16:30
	MNIST own exploration	Team work in break-out rooms	Lecture 07a notebook	15	16:45
	Sentiment analysis intro	Lecture / Q&A	Lecture 07b notebook	15	17:00
	Sentiment analysis	Team work in break-out rooms	Lecture 07b notebook	35	17:35
17:50-19:20	Sentiment analysis results	Lecture / Q&A	Lecture 07b notebook	15	18:05
	Decision trees	Lecture / Q&A	Lecture slides	20	18:25
	Random forests	Lecture / Q&A	Lecture slides	5	18:30
	Gradient Boosted Trees	Lecture / Q&A	Lecture slides	5	18:35
	Cover type example	Lecture / Q&A	Lecture 07c notebook	20	18:55
	Cover type exploration	Team work in break-out rooms	Lecture 07c notebook	25	19:20

Types of problems

Fachhochschule Dortmund
University of Applied Sciences and Arts

Supervised Approaches

- Labeled data
- Target values known

Classification

- Predict category

Regression

- Predict numeric value

Unsupervised Approaches

- Unlabeled data
- No target value provided

Cluster Analysis

- Organize similar cases into segments

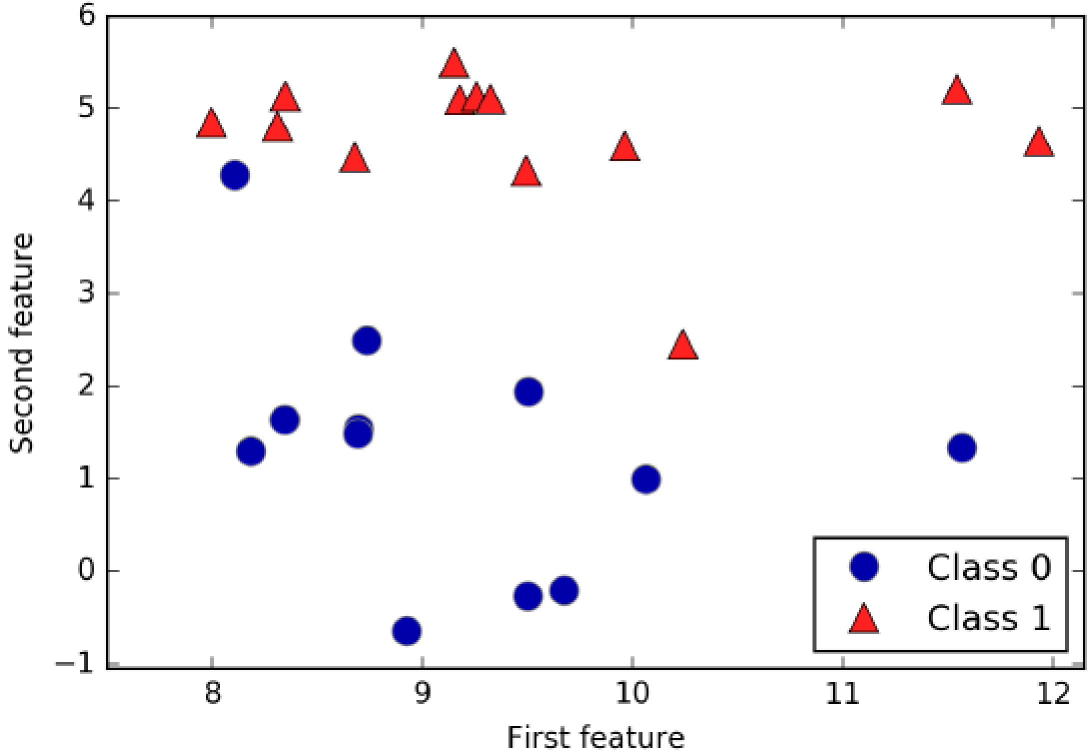
Dimensionality reduction

- Reduce number of features

Question for discussion

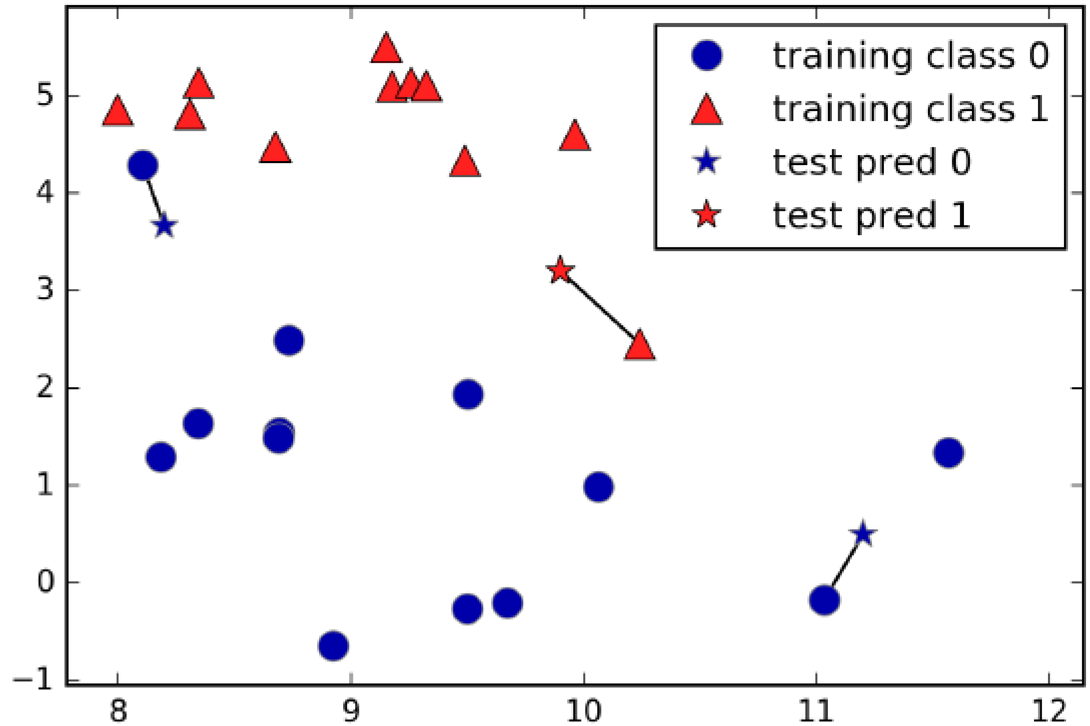
- Find examples for each of the 4 categories

k-Nearest Neighbors – example



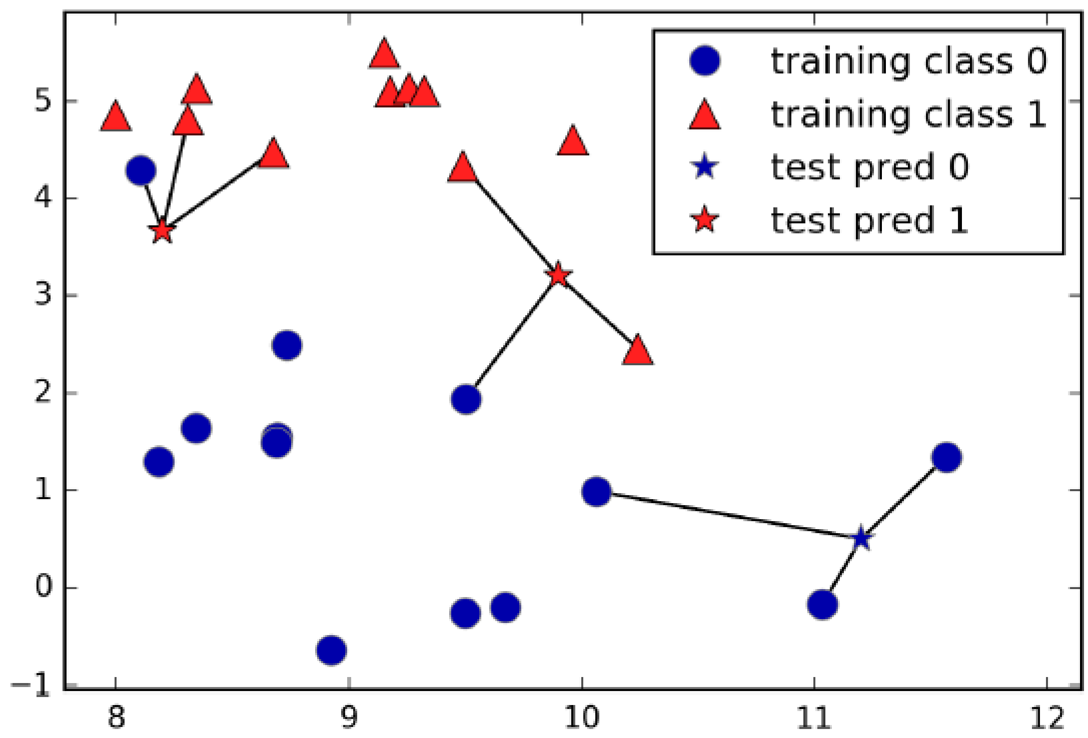
Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

k-Nearest Neighbors – predictions 1-NN



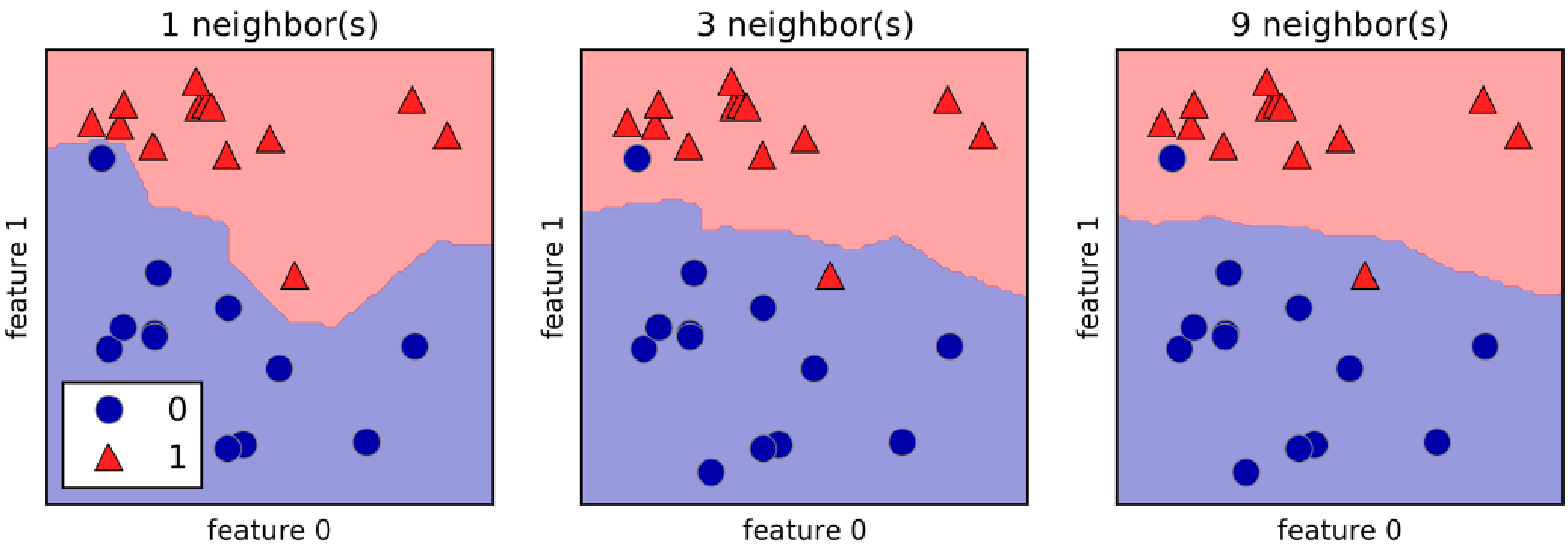
Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

k-Nearest Neighbors – predictions 3-NN



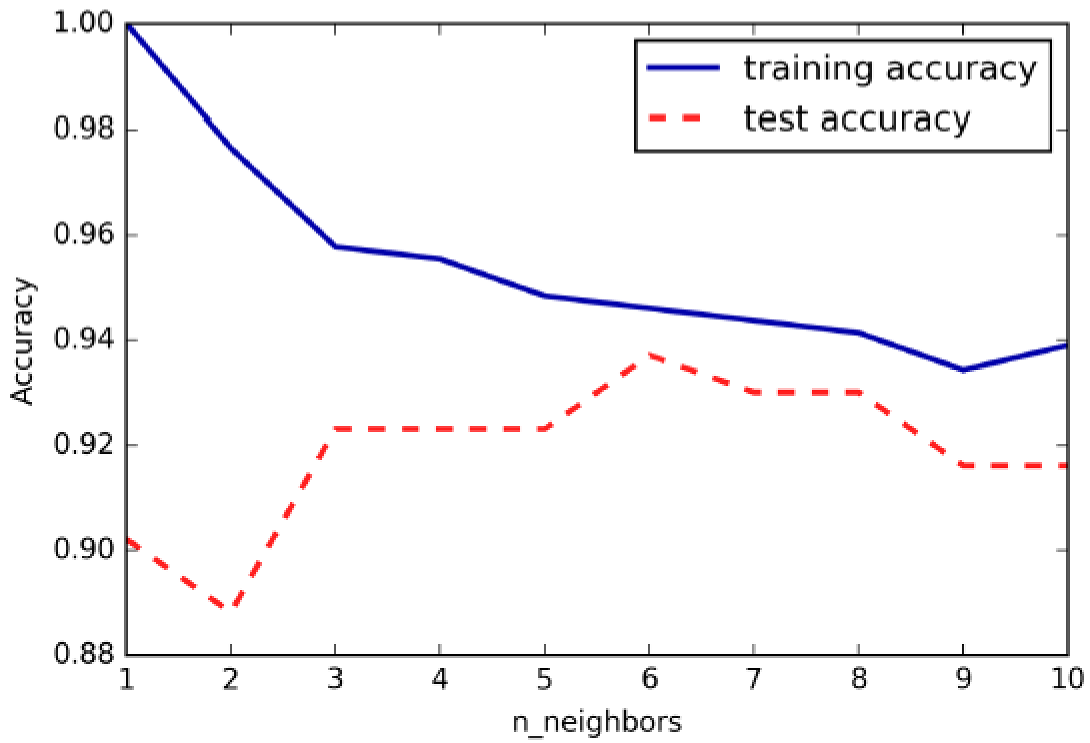
Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

k-Nearest Neighbors – Decision Boundary



Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

k-Nearest Neighbors – Accuracy



Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

k-Nearest Neighbors – Summary

Parameters

- Number of neighbors k
- Distance metric (Euklidean as default)

Strengths

- Easy to understand
- Building model is fast

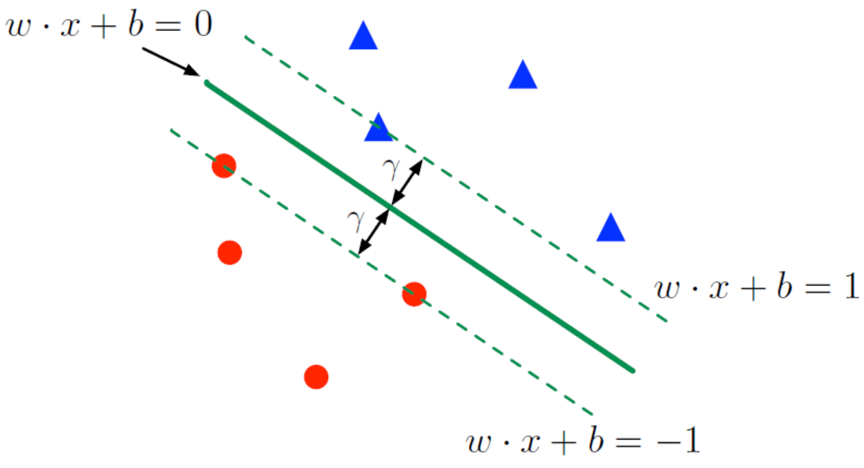
Weaknesses

- Making predictions is very slow on large datasets
- Usually not good with many features (hundreds or more)
- Particularly bad with sparse datasets (many zeros)
- Not robust if features are on different scales

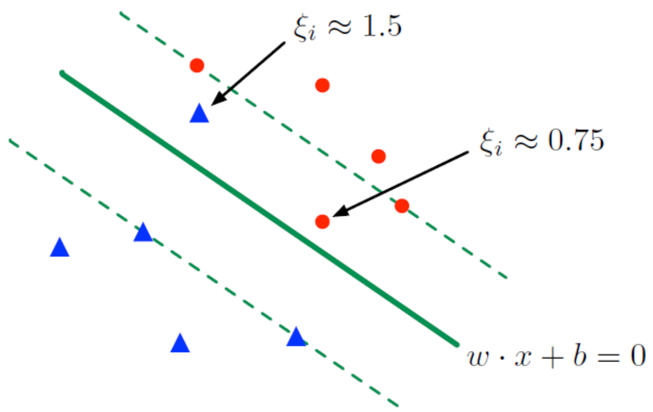
Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Linear Support Vector Machines

Maximize margin $\gamma = \frac{1}{\|\boldsymbol{w}\|}$



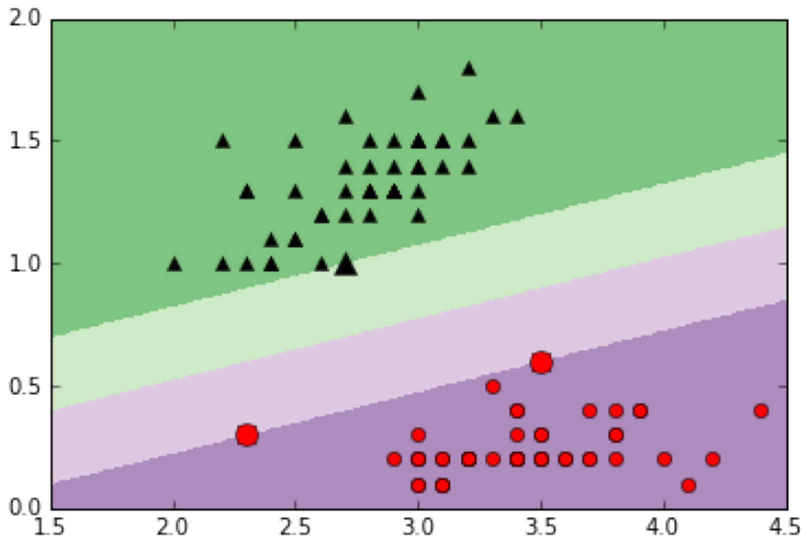
Minimize slack ξ

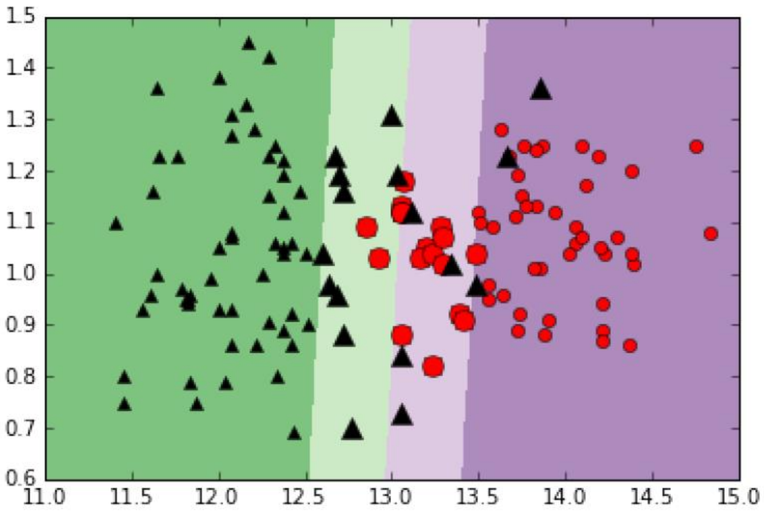


Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Linear Support Vector Machines

Support vectors





Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Linear Binary Classification – Formulas

Given training input data $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^{\boldsymbol{d}}$ with labels $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n)} \in \{-\mathbf{1}, \mathbf{1}\}$.

Try to find $\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}}$ and $\boldsymbol{b} \in \mathbb{R}$, so that

$$\mathbf{sign}\left(\boldsymbol{w} \cdot \boldsymbol{x}^{(i)} + \boldsymbol{b}\right) = \boldsymbol{y}^{(i)}$$

as often as possible.

Logistic regression

Minimize

$$\boldsymbol{L}(\boldsymbol{w}, \boldsymbol{b}) = -\ln\left(\prod_{i=1}^{n} \boldsymbol{Pr}_{\boldsymbol{w},\boldsymbol{b}}\left(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}\right)\right) + \boldsymbol{\lambda}\|\boldsymbol{w}\|_2^2 = -\sum_{i=1}^{n} \ln\left(\frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{y}^{(i)}\left(\boldsymbol{w}\cdot\boldsymbol{x}^{(i)}+\boldsymbol{b}\right)}}\right) + \boldsymbol{\lambda}\|\boldsymbol{w}\|_2^2$$

Linear support vector machines

Hard margin

$$\min_{\boldsymbol{w}\in\mathbb{R}^{\boldsymbol{d}}, \boldsymbol{b}\in\mathbb{R}} \|\boldsymbol{w}\|_2^2 \quad \text{such that} \quad \boldsymbol{y}^{(i)}\left(\boldsymbol{w} \cdot \boldsymbol{x}^{(i)} + \boldsymbol{b}\right) \geq 1$$

Soft margin

$$\min_{\boldsymbol{w}\in\mathbb{R}^{\boldsymbol{d}}, \boldsymbol{b}\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^{\boldsymbol{n}}} \|\boldsymbol{w}\|_2^2 + \boldsymbol{K}\sum_{i=1}^{n} \boldsymbol{\xi}_{\boldsymbol{i}} \quad \text{such that} \quad \boldsymbol{y}^{(i)}\left(\boldsymbol{w} \cdot \boldsymbol{x}^{(i)} + \boldsymbol{b}\right) \geq \mathbf{1} - \boldsymbol{\xi}_{\boldsymbol{i}}$$

$$\boldsymbol{\xi} \geq \mathbf{0}$$

Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Linear Models – Summary

Parameters

- Regularization parameter alpha and C
- Model type lasso vs. ridge for regression / logistic vs. SVM for classification

Strengths

- Fast to train, fast to predict
- Work well with sparse data
- Relatively easy to understand how predictions are made
- Work well with large number of features

Weaknesses

- Coefficients hard to interpret, especially if features are highly correlated
- Sometimes fail with small datasets
- Perform bad with non-linear features and datasets that are not linearly separable

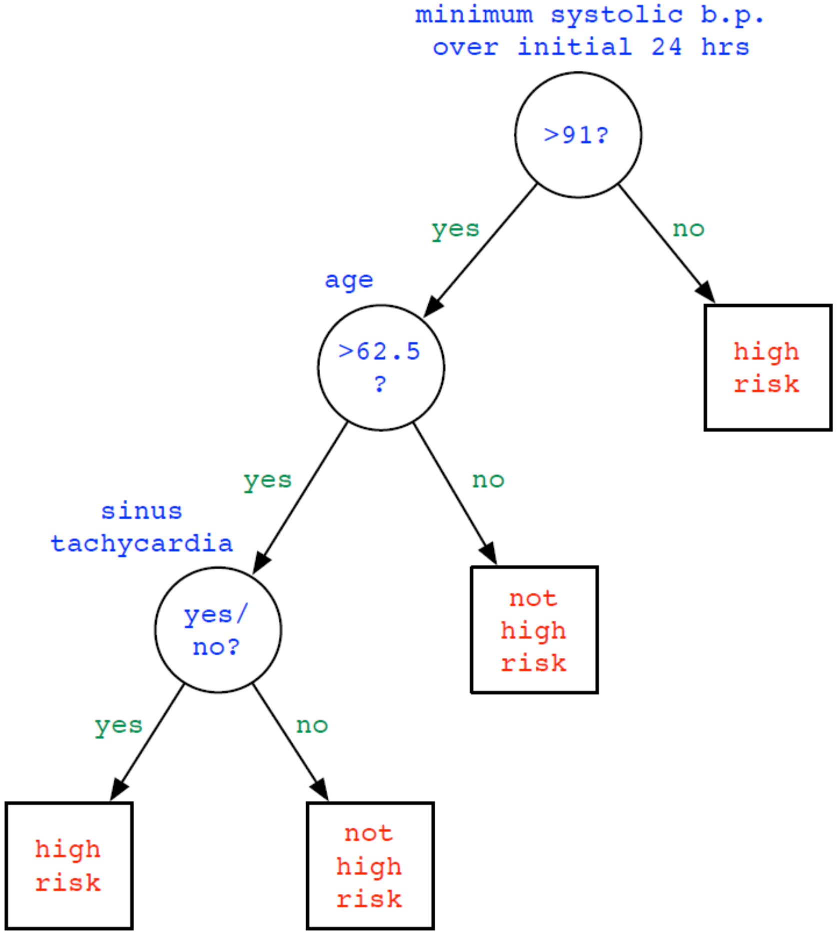
Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Decision Trees

Fachhochschule Dortmund
University of Applied Sciences and Arts

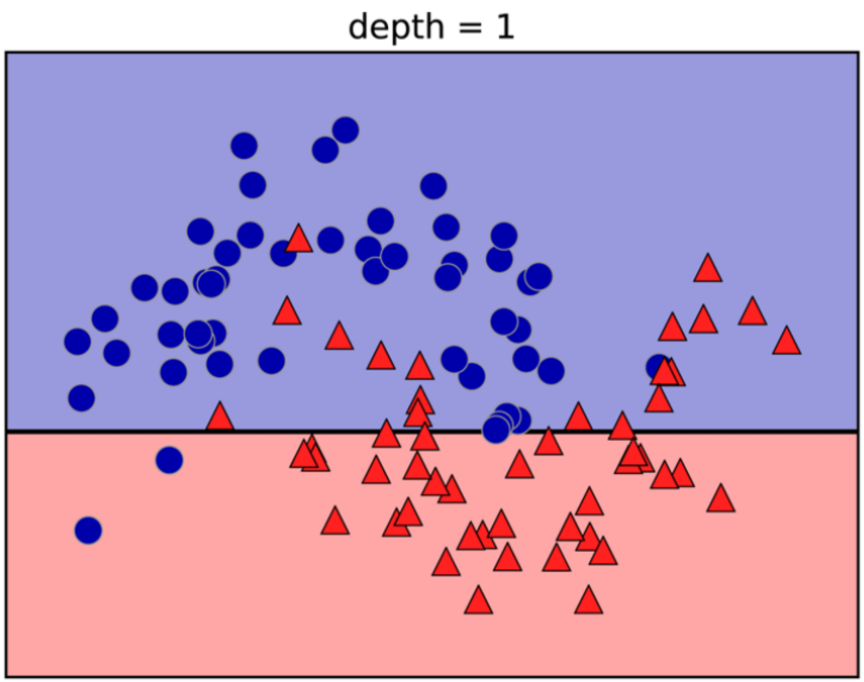
UCSD Medical Center (1970s):
identify patients at risk of dying
within 30 days after heart attack.

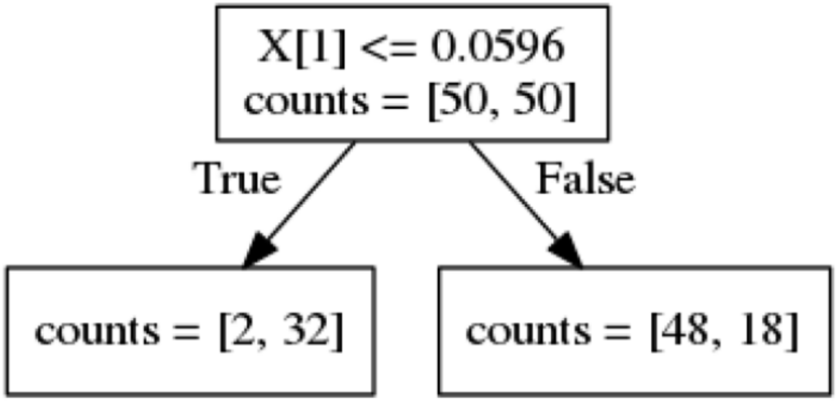
Data set:
215 patients.
37 (=20%) died.
19 features.



Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Complexity of Decision Trees

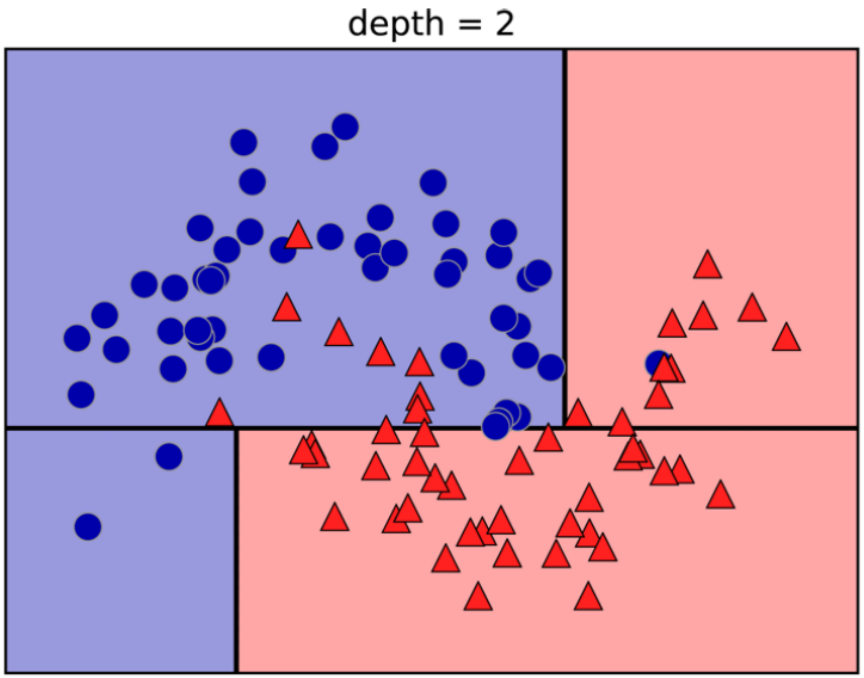


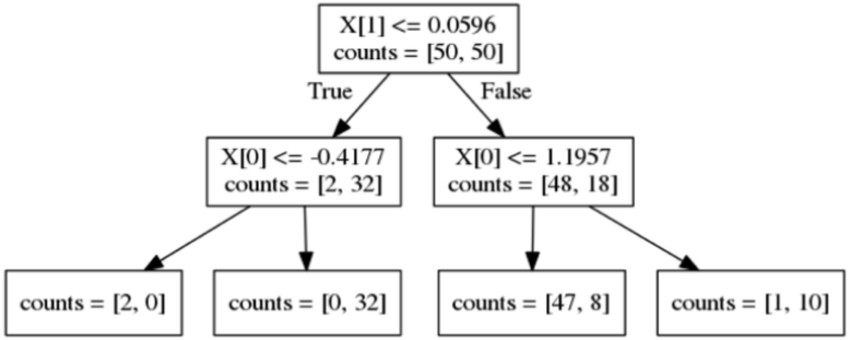


Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Complexity of Decision Trees

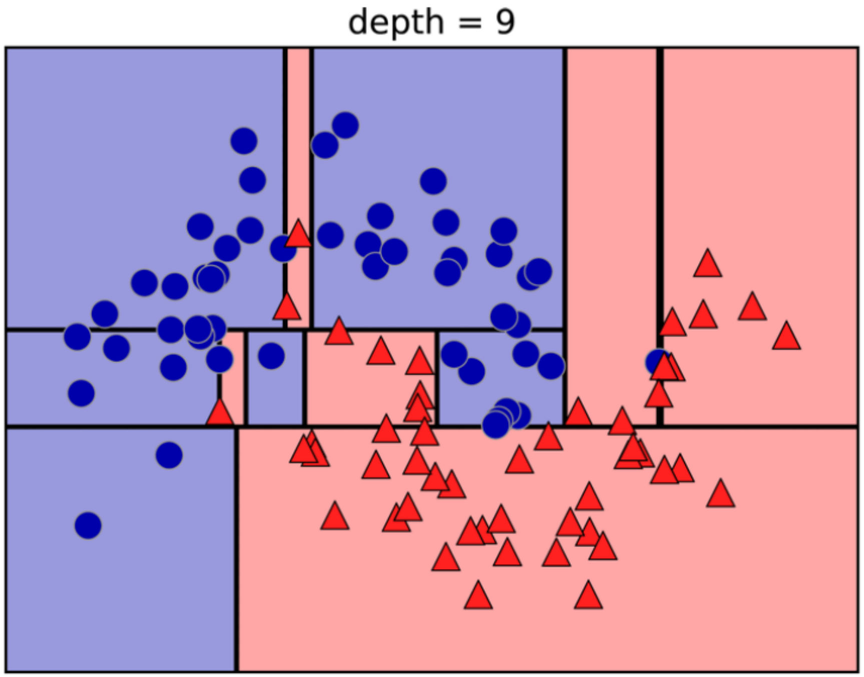
Fachhochschule Dortmund
University of Applied Sciences and Arts

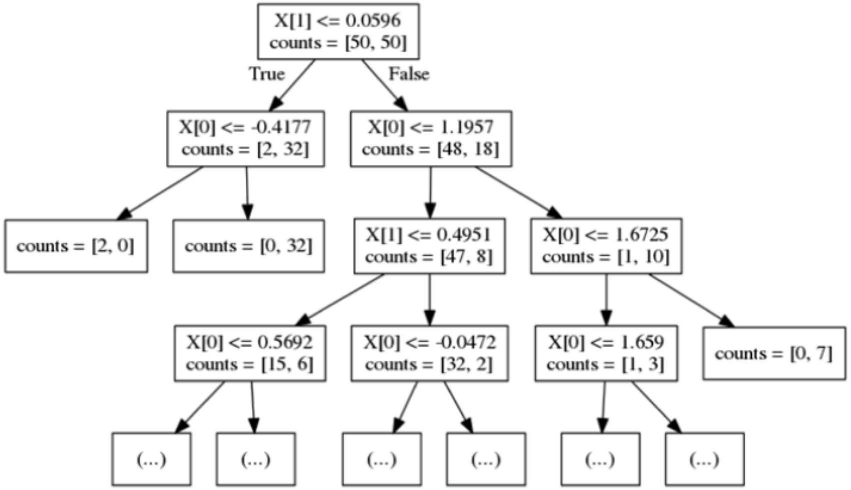




Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Complexity of Decision Trees





Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Decision Trees – Summary

Parameters

- Complexity parameters max_depth, max_leaf_nodes, min_samples_leaf

Strengths

- Easy to visualize and explain to non-experts
- No scaling of data required
- Work with mix of numeric and categorical input variables
- Fast to train, fast to predict

Weaknesses

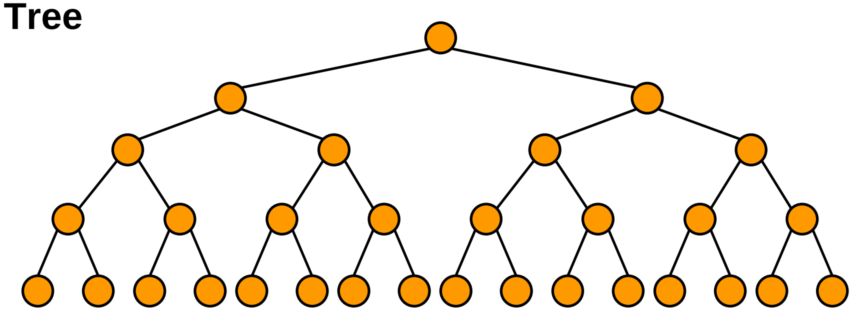
- High risk of overfitting

Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Random forests vs. Gradient Boosted Trees

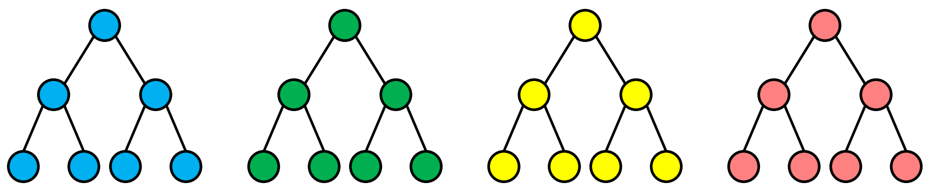
Fachhochschule Dortmund
University of Applied Sciences and Arts

Single Decision Tree



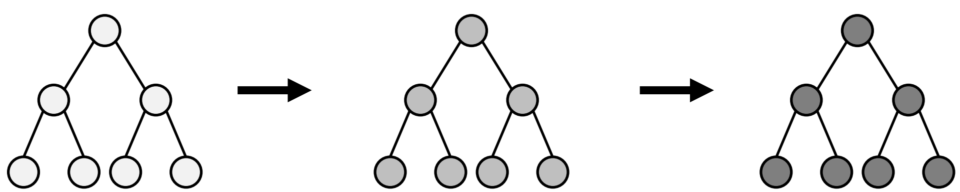
- One (large) tree
- Based on all data points
- Each split based on all features

Random Forest



- Many independent (small) trees
- Each tree based on random subset of datapoints
- Each split based on random subset of features
- Prediction based on majority vote of all trees

Gradient Boosted Trees



- (Small) trees built sequentially
- Try to improve previous tree by weighting falsely predicted data points higher
- Prediction based on weighted vote of all trees

Random Forests – Summary

Fachhochschule Dortmund
University of Applied Sciences and Arts

Parameters

- Number of iterations/trees n_estimators (the more the better)
- n_jobs = -1 to use all available cores will speed up the model
- Complexity parameters max_depth, max_features (sqrt(n_features) by default), max_leaf_nodes

Strengths

- Very powerful, widely used in modern machine learning
- No scaling of data required
- Work with mix of numeric and categorical input variables
- Faster than boosted trees, because trees can be built in parallel
- Less danger of overfitting compared to single decision trees and boosted trees

Weaknesses

- Interpretation and explanation of results very difficult in contrast to single decision trees
- Usually don’t work well on sparse data (like text data)
- On large datasets slower than linear models

Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Gradient Boosted Trees – Summary

Fachhochschule Dortmund
University of Applied Sciences and Arts

Parameters

- Number of iterations/trees n_estimators (higher number might lead to overfitting)
- learning_rate: To what extent the model is allowed to correct the errors of the previous model
- Complexity parameters max_depth, max_leaf_nodes, min_samples_leaf

Strengths

- Very powerful, widely used in modern machine learning
- No scaling of data required
- Work with mix of numeric and categorical input variables
- Less danger of overfitting compared to single decision trees

Weaknesses

- Time-consuming to train compared to random forests, because trees have to be built sequentially
- Interpretation and explanation of results very difficult in contrast to single decision trees

Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media