



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

2024 FS CAS PML - Supervised Learning

2 Klassifikation

2.5 Multiklass Klassifikatoren

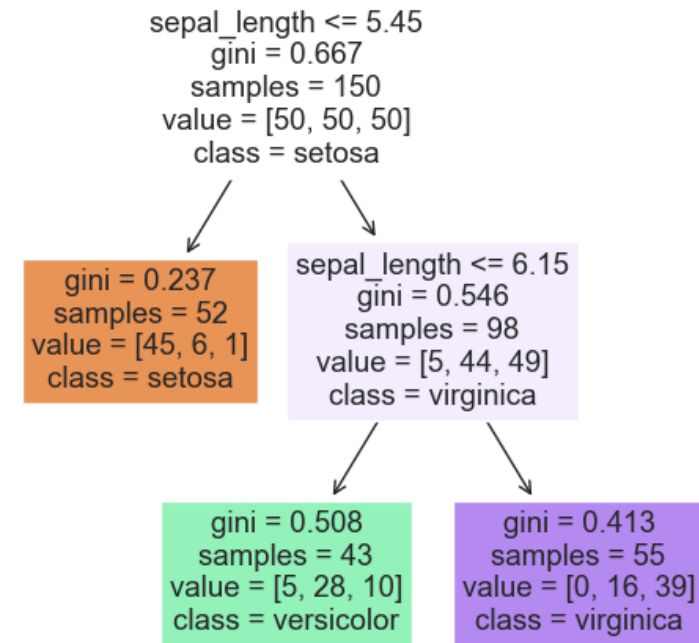
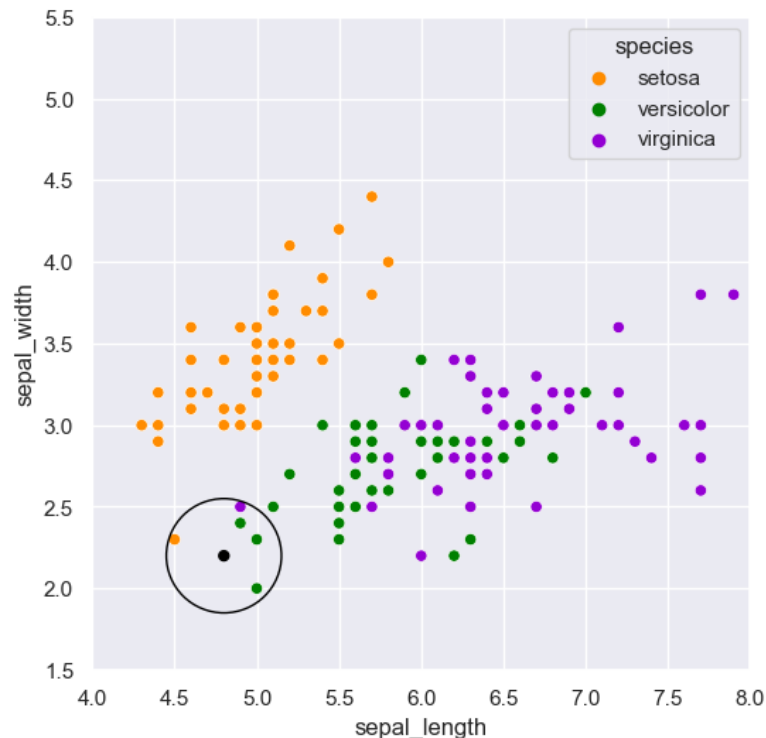
Werner Dähler 2024

2 Klassifikation - AGENDA

- 21. Instanzbasierte Modelle
- 22. Regelbasierte Modelle
- 23. Mathematische Modelle
- 24. Neuronale Netze
- 25. **Multiklass Klassifikation**

2.5 Klassifikation - Multiklass Klassifikation

- ▶ alle bisher vorgestellten Klassifikatoren können auch für Multiklass Fragestellungen eingesetzt werden, d.h. das Target weist eine Kardinalität >2 auf
- ▶ bei einigen ist es offensichtlich
 - ▶ Instanzbasierte Methoden (links)
 - ▶ Regelbasierte Methoden (rechts)



2.5 Klassifikation - Multiklass Klassifikation



- ▶ andere Klassifikatoren bauen im Kern auf Algorithmen auf, welche ausschliesslich für binäre Klassifikation konzipiert sind, z.B.
 - ▶ Support Vektor Maschinen
 - ▶ Logistische Regression
 - ▶ Neuronale Netze
- ▶ um diese trotzdem für Multiklass Fragestellungen einsetzen zu können, stehen in scikit-learn entsprechende Multiklass Strategien zur Verfügung
- ▶ die zwei wichtigsten sind:
 - ▶ One-vs-Rest
 - ▶ One-vs-One
- ▶ ausserdem Parameter, welche die Strategien steuern, sind z.B.
 - ▶ `decision_function_shape` bei SVC
 - ▶ `multi_class` bei LogisticRegression

2.5 Klassifikation - Multiklass Klassifikation



One-vs-Rest

- ▶ für jede Klasse des Targets wird ein unabhängiges Modell trainiert
- ▶ dabei wird jeweils die fragliche Klasse mit 1 und alle andern mit 0 codiert, im Falle von iris:
 - ▶ 1: setosa, 0: versicolor und virginica
 - ▶ 1: versicolor, 0: setosa und virginica
 - ▶ 1: virginica, 0: setosa und versicolor
- ▶ für die Prediction werden von allen Modellen die Wahrscheinlichkeiten für den Targetwert 1 ermittelt (vgl. `predict_proba()`) und die Klasse mit dem höchsten Wert zurückgegeben

2.5 Klassifikation - Multiklass Klassifikation



One-vs-One

- ▶ für jedes Paar von Klassen wird ein binäres Modell trainiert, im Fall von iris z.B.
 - ▶ 1: setosa, 0: versicolor
 - ▶ 1: setosa, 0: virginica
 - ▶ 1: versicolor, 0: virginica
- ▶ d.h. für n Klassen werden $\frac{n(n-1)}{2}$ Modelle trainiert
- ▶ für die Prediction werden wiederum von allen Modellen die Wahrscheinlichkeiten für den Targetwert 1 ermittelt und die Klasse mit dem höchsten Wert zurückgegeben
- ▶ Vergleich vs. One-vs-Rest:
 - ▶ erstellt bei hoher Kardinalität des Targets viel mehr Modelle, z.B. bei 10 Klassen 45!
 - ▶ andererseits wird für das Trainieren der einzelnen Modelle jeweils nur ein Teil der Beobachtungen berücksichtigt, was das Ganze wiederum beschleunigt

2.5 Klassifikation - Multiklass Klassifikation



- ▶ scikit-learn weist verschiedene Meta Learner auf, welche die genannten Strategien unterstützen, z.B.
 - ▶ [multiclass.OneVsRestClassifier](#)
 - ▶ [multiclass.OneVsOneClassifier](#)
- ▶ diese benötigen einen Basis Klassifikator, sind aber eher für experimentelle Arbeiten gedacht
- ▶ vgl. [ipynb]

2 Klassifikation

Workshop 07

Gruppen zu 2 bis 4, Zeit: 30'

- ▶ vergleichen Sie alle bis jetzt vorgestellten Klassifikatoren miteinander in Bezug auf
 - ▶ Performance
 - ▶ Rechenzeit, differenziert nach `.fit()` und `.predict()` und visualisieren Sie die Ergebnisse
- ▶ Tipp: modifizieren / ergänzen Sie dazu den abgegebenen Code von Kapitel 2.2.7 Modellvergleiche
- ▶ optional: fügen Sie andere, im Kurs nicht behandelte Klassifikatoren dazu, welche Sie in der Dokumentation zu scikit-learn finden
- ▶ optional: falls Sie im Rahmen von Feature Engineering alternatives Preprocessing erarbeitet haben, können Sie die Auswirkungen desselben jetzt auch noch einbeziehen
- ▶ optional: wie wirkt sich Skalierung (z.B. mit `StandardScaler`) auf die Performance von `MLPClassifier` aus?

