

TECHNICAL APPLICATIONS AND DATA MANAGEMENT. WS 2021/2022.

VORLESUNG 6

19.10.2021

MÜNCHEN

STUDIENGANG
DIGITAL
MANAGEMENT.



AGENDA

1. Überwachtes Lernen:

- Workflow
- „Simple“ Algorithmen
- Kombinierte Algorithmen/ Ensemble Learning
- Metriken für überwachtes Lernen

2. Case Study

WAS HABEN WIR BIS JETZT GEMACHT?

ROADMAP	WAS HABEN WIR GEMACHT?
Vorlesung 1	Workflow Data Management, Datentypen und Datenqualität
Vorlesung 2	Einführung Data Science und Data Science Workflow, Grundlagen Data Management
Vorlesung 3	Grundlagen Stochastik: Wahrscheinlichkeitsrechnung, deskriptive und explorative Statistik
Vorlesung 4	Statistische Inferenz, lineare Regression
Vorlesung 5	Einführung Machine Learning, Unüberwachtes Lernen

1. ÜBERWACHTES LERNEN

SUPERVISED LEARNING: ÜBERSICHT.

Ziel: Wir suchen ein Modell um für beliebige **Eingaben $X_{i,...,n}$** eine **Zielgröße Y** möglichst genau zu bestimmen oder anzunähern.

Ermöglicht:

- **Regression:** Vorhersage eines kontinuierlichen Wertes Y (bspw. natürliche oder reelle Zahl).
- **Klassifikation:** Vorhersage eines diskreten Wertes Y (bspw. wahr/ falsch, Tierart, Personenentdeckung,).

Was schauen wir uns?

- Gesamter Workflow: von Datenaufbereitung, Trainieren, Testen bis hin zur Anwendung.
- Ausgewählte „simple“ Algorithmen für Regression und Klassifikation (weitere Algorithmen im Backup).
- Kombinierte Algorithmen (Ensemble Learning).
- Metriken.

SUPERVISED LEARNING: REGRESSION.

Ziel: Vorhersage eines Wertes aus einer großen Wertemenge (kontinuierliche Variable).

Ermöglicht: Prädiktion eines Wertes

Aber:

- Hoher Aufwand für das Bestimmen der Zielwerte (Labeln). Dies ist beim unüberwachten Lernen nicht vorhanden.
- Ist ungenauer als Klassifikation (wir versuchen einen genauen Wert aus einer großen Wertemenge vorherzusagen).
- Oft wird diese zusätzliche "Genauigkeit" nicht benötigt; z.B. reicht im Aktienhandel oft Prädiktion, ob Kursziel überschritten.

Anwendungsbeispiele:

- Aktienkursvorhersage: Prädiktion eines genauen Kurswertes.
- Biomonitoring: Prädiktion Cholesterin-/ Insulinspiegel.
- ...

SUPERVISED LEARNING: KLASSIFIKATION

Ziel: Vorhersage eines Wertes aus einer kleineren, abzählbaren Menge (diskrete Variable).

Ermöglicht:

- Klassifizieren: Einordnen eines Wertes in eine endliche Gruppe bzw. Prädiktion eines endlichen Wertes.

Aber:

- Hoher Aufwand für das Bestimmen der Zielwerte (Labeln). Dies ist beim unüberwachten Lernen nicht vorhanden.
- Genauigkeit hängt von der Größe der Gruppe ab (für sehr große Gruppen haben wir quasi eine Regression).

Anwendungsbeispiele:

- Online-Werbung: klickt der User die Werbung an oder nicht an?
- Bilderkennung: Erkennen eines bestimmten Tieres, Schriftzeichen, Bildinhalte bewerten, ...
- Übersetzung: Deutsch zu Spanisch, ...
- ...



2. GENERISCHER ABLAUF ÜBERWACHTES LERNEN

GENERISCHER ABLAUF SUPERVISED LEARNING. ÜBERSICHT.

1. Daten organisieren und hochladen.
2. Daten aufbereiten/ Data cleaning.
3. Daten aufteilen in Test- und Trainingsmenge (sowie ggf. Validierungsmenge).
4. Vorbereitungen: Machine Learning-Algorithmus und Kostenfunktion wählen sowie Modellparameter initialisieren.
5. Training: schrittweise Optimierung Modellparameter bis Modell möglichst gute Performance für die Trainingsmenge hat.
6. Modell(-güte) validieren anhand der Testmenge. Falls Modellgüte in Ordnung ist, weiter zu Schritt 7. Sonst zu Schritt 5.
7. Deployment: Modell einsetzen im „Live“-Betrieb inkl. Kontinuierliches Überprüfen der Modellgüte und ggf. Aktualisierung.

Das haben wir uns schon in
vorigen Vorlesungen angesehen

Ziel: Lernen eines möglichst genauen Modells $H(\text{Input}) = \text{Zielgröße}^1$

VERANSCHAULICHUNG GENERISCHER ABLAUF MACHINE LEARNING ANHAND DES IRIS-DATENSATZ.

- Fisher, „The use of multiple measurements in taxonomic problems” (1936).
- Datenset enthält 50 Samples à 4 Features zu jeder der 3 Irissorten:
 - Sepal length (Kelchblattlänge)
 - Sepal width (Kelchblattweite)
 - Petal length (Blütenblattlänge)
 - Petal Width (Blütenblattweite)
- Datensatz wird sehr häufig als Testdatensatz für Klassifikation verwendet.
- Aber: kleiner Datensatz

Aufgabenstellung: Bestimmen Klasse der Iris anhand Messungen der 4 Features

Zielgrösse: Klasse der Iris



Iris Setosa
(Klasse 0)



Iris versicolor
(Klasse 1)



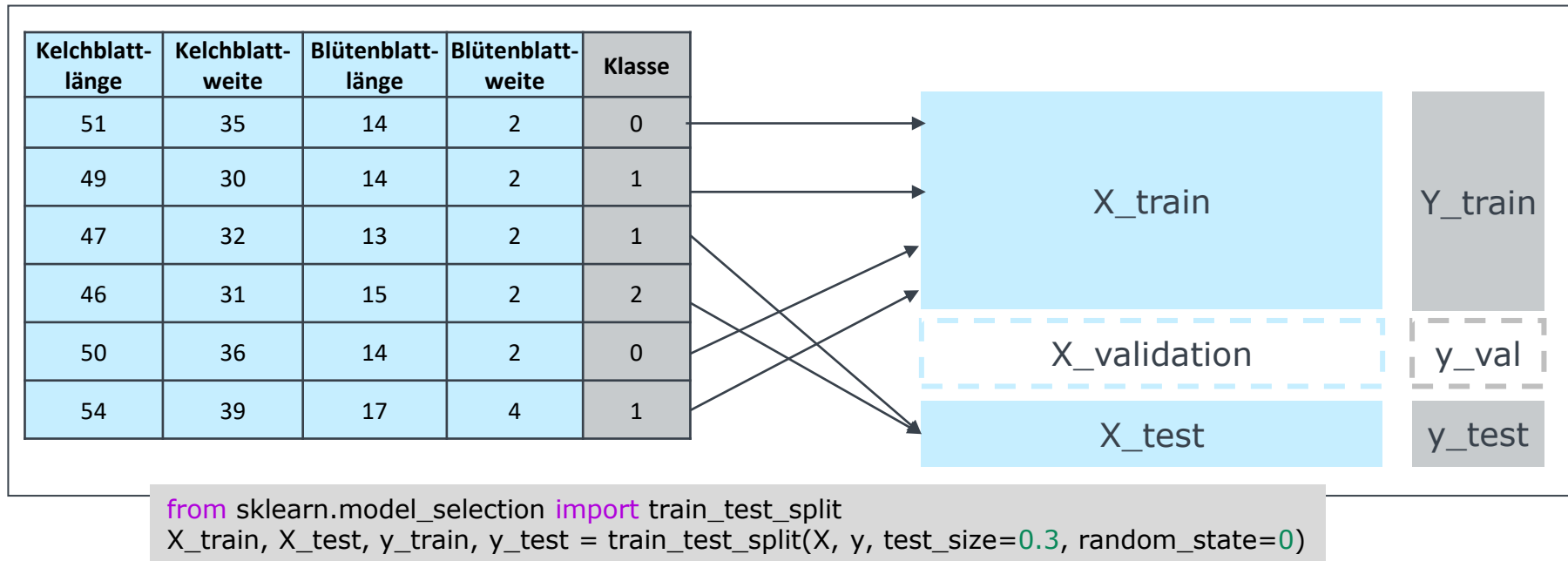
Iris virginica
(Klasse 2)

GENERISCHER ABLAUF SUPERVISED LEARNING. SCHRITT 1 UND SCHRITT 2: DATEN ORGANISIEREN UND AUFBEREITEN.

Hatten wir bereits in der 2. Vorlesung

GENERISCHER ABLAUF SUPERVISED LEARNING. SCHRITT 3: DATENAUFTEILUNG.

Matrix mit den Features/ Eigenschaften
Vektor mit den Zielwerten



- Aufgabe **Trainingsset**: Lernen/ Optimieren der Parameter des Machine Learning Modells.
- Aufgabe **Testset**: Evaluation, wie gut die Parameter und somit das Modell sind.
- **Zufällige Aufteilung** in Trainings- und Testmenge stellt ähnliche statistische Eigenschaften sicher (sortierte Ausgangsdaten!)
- Oft wird **Validierungsset** zur Optimierung der Parameter des Trainingsprozesses (**Hyperparameter**) gebildet.
- Verhältnis Trainings- zu Testmenge: oft 70% zu 30%. Bei Einsatz Validierungsmenge: 60%-20%-20%.

GENERISCHER ABLAUF SUPERVISED LEARNING.

SCHRITT 4: MODELL WÄHLEN UND INITIALISIEREN

1. Wahl Machine Learning-Modell abhängig von:

- Datentyp: Zahlen, Bilder, Sprache,
- Verfügbaren Ressourcen: Rechen-Power, Zeit, Kosten, ...
- Datenmenge: viele oder wenig Daten vorhanden?
- Qualität Daten: fehlende Werte/ Ausreißer stören manche Lernverfahren.
- Anforderungen an Modellgüte.
- Komplexität Algorithmus: komplexe Algorithmen neigen zu hoher Streuung, simple zu hoher Verzerrung.

Bias-/ Variance
Tradeoff

2. Initialisierung Modellparameter

- Hyperparameter: Parameter Trainingsprozess, bspw. Learning rate Neural Network oder Anzahl Cluster K-Nearest Neighbours
- Modellparameter (Weights w_i und Bias b): Initialisierung mit zufälligen Werten (für schnelleres Lernen).

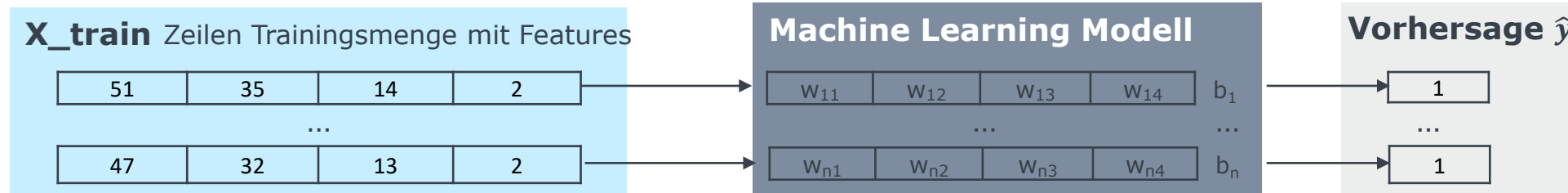
3. Wahl Loss-Funktion: Bewertungsfunktion für Genauigkeit des Machine Learning Modells (bspw. MSE, Cross-Entropy¹, ...)

Auswahl ist erfahrungsbasierend. Die meisten Machine Learning Bibliotheken treffen aber standardmäßig gute Vorauswahlen.

GENERISCHER ABLAUF SUPERVISED LEARNING.

SCHRITT 5: TRAINING

1. Feed Forward: Vorhersage Wert je **Sample** (individuelle Zeile Datensatz)



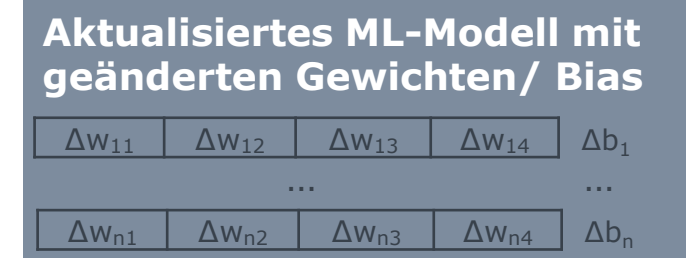
2. Berechnen Modellgüte und Fehler des Modells

- **Loss Function:** Berechnen Delta zwischen realem und vorhergesagtem Wert je Sample.
- **Cost Function:** Durchschnitt der Losses für jedes Sample.



3. Minimierung Fehler und Anpassen Modellparameter:

- Aufstellen Gleichung für Minimierung Cost Function (erste Ableitung gleich 0 setzen).
 - Lösen Gleichung für die Modellparameter.
 - Aktualisieren Modellparameter (Δw_{ij} und Δb_i), um so im nächsten Trainingslauf näher am realen Y-Wert zu landen.
- Einsatz iterativer Algorithmus bei hochdimensionalen Daten (bspw. Gradient Descent).

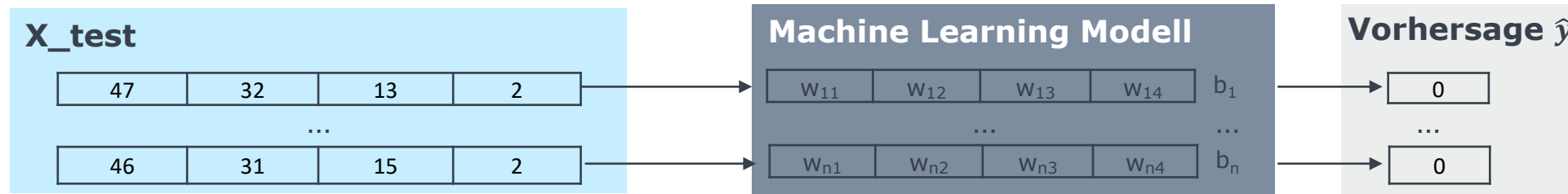


Durchführen Schritte 1-3 inkl. Update Modellparameter für gesamtes Trainingsset heißt **Epoch**, für Teile Trainingsset **Batch**. Anzahl Durchläufe je Batch/ Epoch erfahrungsabhängig inkl. Abbruchkriterien (ausreichende oder stagnierende Güte)

GENERISCHER ABLAUF SUPERVISED LEARNING.

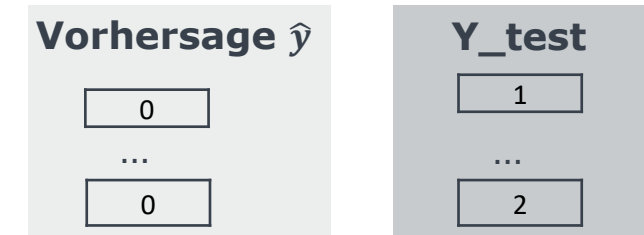
SCHRITT 6: MODELL TESTEN.

1. Feed Forward: Vorhersage Wert je **Sample**, d.h. eine individuelle Zeile des Datensatzes



2. Berechnen Modellgüte und Fehler des Modells

- **Validation Accuracy:** Berechnen Delta zwischen realem und vorhergesagtem Wert
- **Klassifikationsmetriken:** Confusion Matrix
- **Regressionsmetriken:** MSE, RSME, ...



GENERISCHER ABLAUF SUPERVISED LEARNING.

SCHRITT 6: MODELL TESTEN – WIE BEWERTET MAN DIE GÜTE EINES MODELLS?

Klassifizierungsmetriken

Confusion Matrix

		Vorhergesagt	
		Ja	Nein
Tatsächlich	Ja	True Positive	False Negative
	Nein	False Positive	True Negative

```
from sklearn.metrics import confusion_matrix
confusion_matrix(real_y, predicted_y)
```

Accuracy

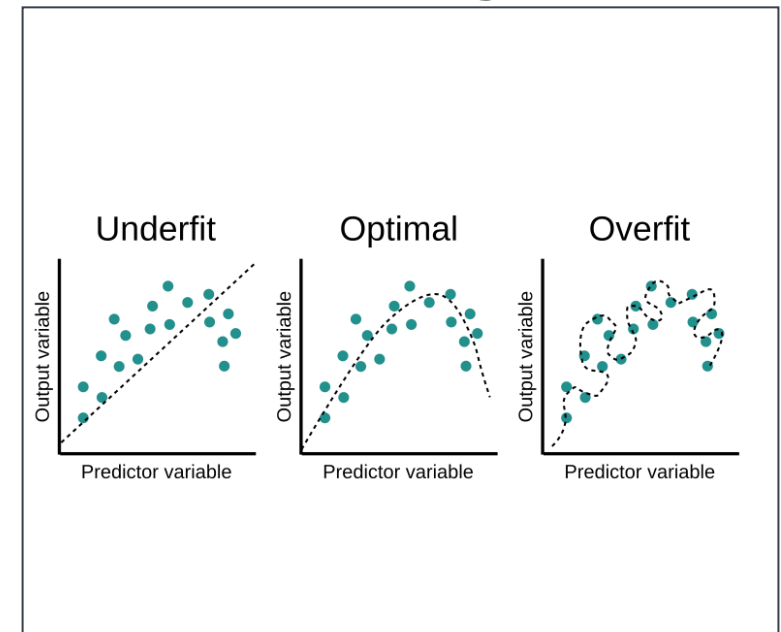
$$\frac{\text{Anzahl richtige Vorhersagen}}{\text{Anzahl aller Vorhersagen}}$$

```
from sklearn.metrics import accuracy_score
accuracy_score(predicted_y, real_y)
```

Regressionsmetriken

- MAE (Mean absolute error)
- MSE
- RSME
- R-squared
- Adjusted R squared
- Explained Variance

Over- und Underfitting



DETAILLIERUNG KLASSIFIZIERUNGSMETRIKEN: CONFUSION MATRIX.

		Vorhergesagt	
		Ja	Nein
Tatsächlich	Ja	True Positive (TP)	False Negative (FN)
	Nein	False Positive (FP)	True Negative (TN)

- **Recall** = wie viele Elemente wurden korrekt vorhergesagt

$$:= \left(\frac{\text{True positive}}{\text{True Positive} + \text{False negative}} \right)$$
- **Specifity** = wie viele negative Element korrekt als negativ vorhergesagt wurden

$$:= \left(\frac{\text{True negative}}{\text{True negative} + \text{False positive}} \right)$$
- **Precision** = wie viele der als "wahr" vorhergesagten Elementen waren wirklich wahr

$$:= \left(\frac{\text{True positive}}{\text{True Positive} + \text{False positive}} \right)$$
- **Accuracy** = die Anzahl korrekt vorhergesagter Elemente geteilt durch die Gesamtzahl

$$:= \left(\frac{\text{True positive} + \text{True negative}}{\text{Gesamtzahl}} \right)$$
- **F1-Score** = je höher der F1-Score, desto besser kann das Modell vorhersagen

$$:= \left(\frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \right)$$

DETAILLIERUNG REGRESSIONSMETRIKEN

- **MAE (Mean absolute error)**: gemittelte Abweichung des prädizierten vom realen Wert. Gut für Feintuning.
→ je kleiner, desto besser das Modell.

$$\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

- **MSE (Mean Squared Error)**: gibt an, wie weit die Prognosewerte um den erwarteten/ realen Wert streut. Durch Quadrierung werden starke Abweichungen besonders "bestraft".
→ Je größer, desto schlechter das Modell.

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

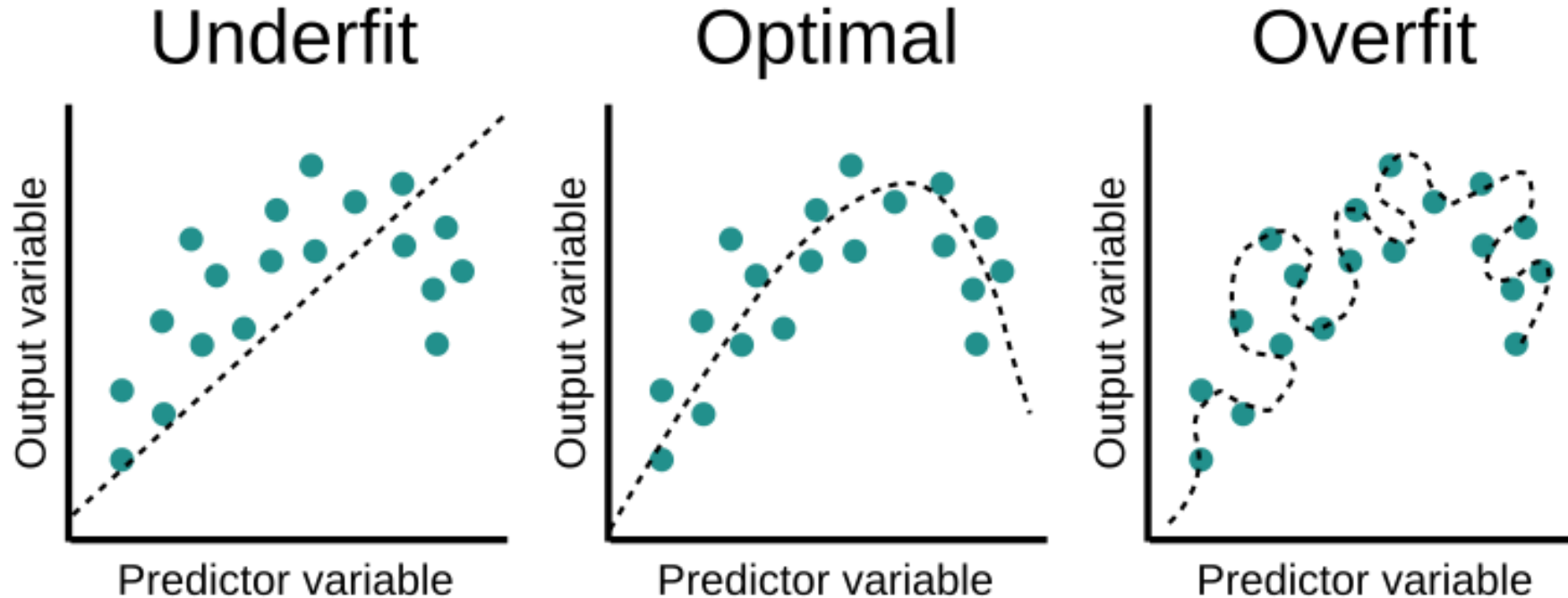
- **RSME (Rooted Mean Squared Error)**: Wurzel aus Mean Squared Error.
→ Je größer, desto schlechter Modell.

$$\sqrt{MSE}$$

- **R-squared** = gibt an, wie gut das Modell die Daten erklären kann; d.h. wie gut das Modell die Y-Werte abdeckt. Wert ist zwischen 0 und 1
→ Wert von 1 bedeutet perfekte Abdeckung, jeder reale Wert wird durch prädizierten Wert abgedeckt.

$$\frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

DETAILLIERUNG OVER- UND UNDERFIT.



Underfitting

Hoher bias/ Verzerrung: wichtige Abhängigkeiten werden nicht erkannt, da Modell zu simpel ist.
→ Erkennbar an: schlechter Accuracy Trainings- und Testdaten.

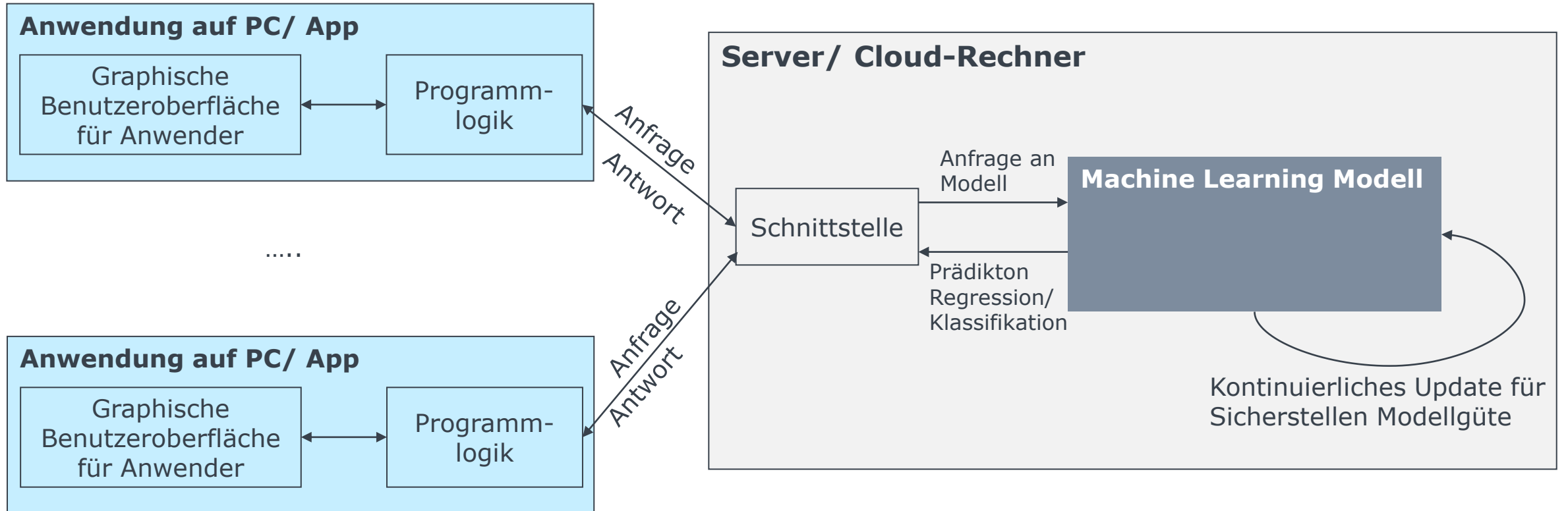
Overfitting

Hohe Varianz: nicht relevante Eigenheiten Trainingsdaten werden gelernt („Auswendiglernen“), Modell zu komplex
→ Erkennbar an: Trainings-Accuracy steigt und ist höher als die Test-Accuracy, Test Accuracy stagniert.

Eselsbrücke anhand Prüfungsvorbereitung:

- Lernten Sie zu wenig, dann haben Sie sowohl bei Übungsprüfungen als auch bei realer Prüfung viele Fehler (Sie wissen nicht was wichtig ist).
- Lernten Sie sehr viel anhand von Übungsprüfungen, dann hilft das nur, falls die der realen Prüfung ähneln (Sie lernten nicht was dran kam).

GENERISCHER ABLAUF SUPERVISED LEARNING. SCHRITT 7: MODELL EINSETZEN (DEPLOYMENT).





3. DETAILLIERUNG AUSGEWÄHLTE ALGORITHMEN

DETAILLIERUNG ALGORITHMEN SUPERVISED LEARNING.

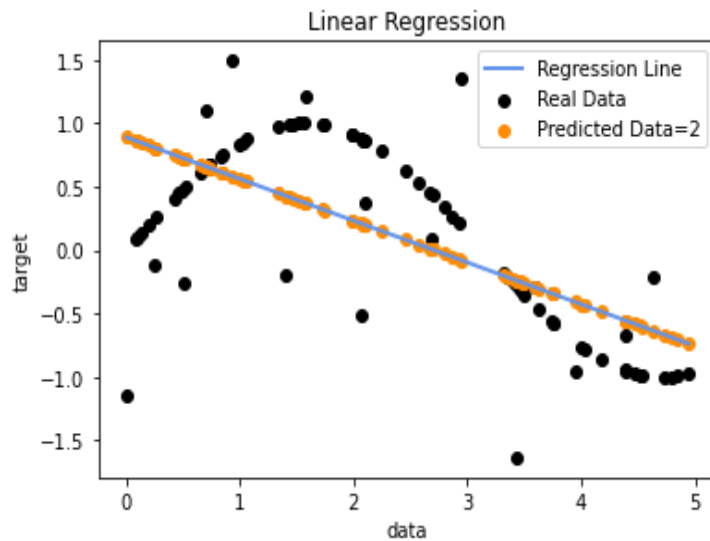
Auf den folgenden Folien werden geläufige Algorithmen an zwei Beispielen: veranschaulicht.

- Regression: Vorhersage der Werte einer Sinuskurve. Dabei ist die Sinuskurve an bestimmten Stellen manuell geändert, um die Vorhersage zu erschweren.
- Klassifikation: anhand des Beispiels Iris-Datensatz. Da dieser Datensatz 4 Features/ Dimensionen hat, läßt er sich schwer graphisch darstellen. Zur Veranschaulichung wurde der Datensatz deshalb auf die 2 Sepal-Werte reduziert.

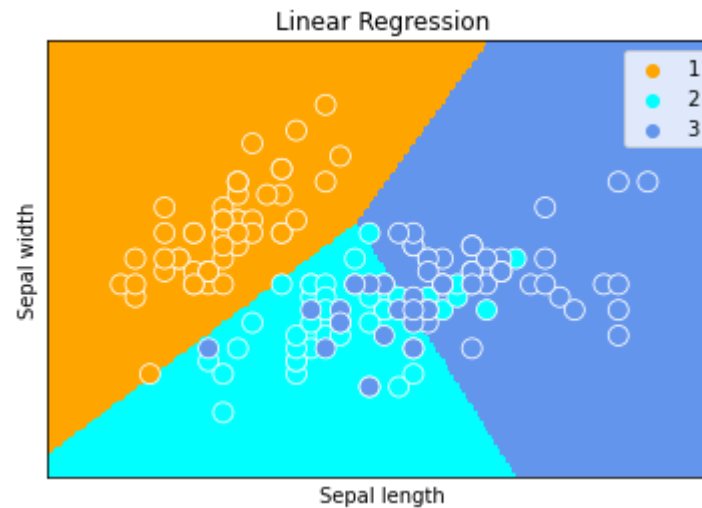
Eine Übersicht bekannter Algorithmen finden Sie bspw. hier: [Link](#)

SUPERVISED LEARNING: LINEARE REGRESSION.

Regression kontinuierliche Variable



Klassifizierung diskrete Variable



Vorteile:

- Leicht verständlicher Algorithmus und Ergebnisse.
- Leicht implementierbar.

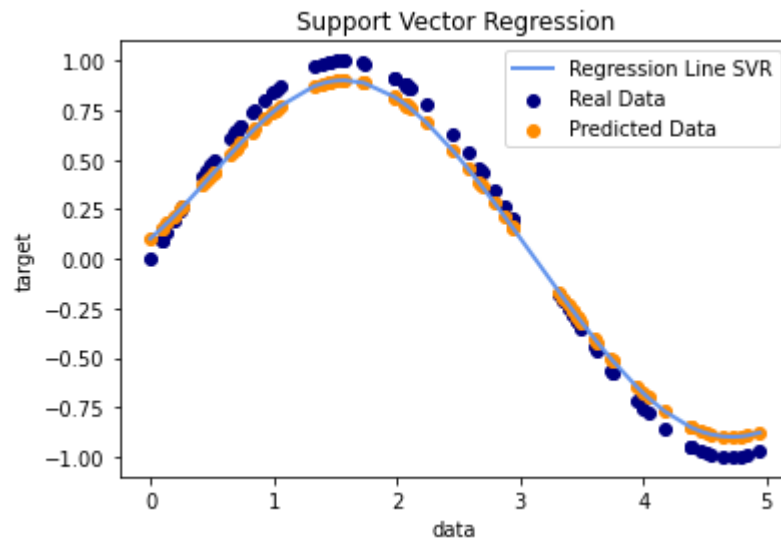
Nachteile:

- Probleme mit Ausreißern.
- Benötigt Zusammenhang/ Abhängigkeit zwischen Variablen.
- Probleme mit komplexen/ hochdimensionalen Daten.

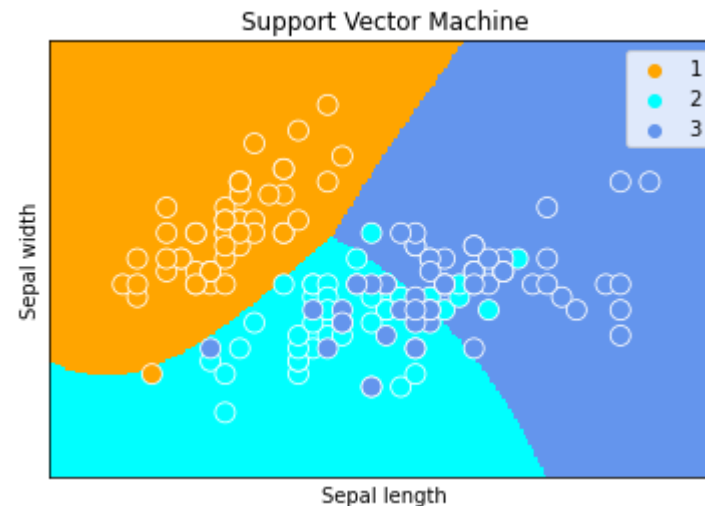
Lineare Regression versucht, die Zielvariable durch ein lineares Modell (Gerade) zu beschreiben. Die Gerade/n ist/sind dabei so konstruiert, daß sie möglichst geringen Abstand von den Punkten des Datensatzes hat.

SUPERVISED LEARNING: SUPPORT VECTOR MACHINE.

Regression kontinuierliche Variable



Klassifizierung diskrete Variable



Vorteile:

- Robust auch bei hochdimensionalen Features.
- Einsetzbar für Klassifikation und Regression.
- Einsetzbar auch für Klassifikation Bilder.

Nachteile:

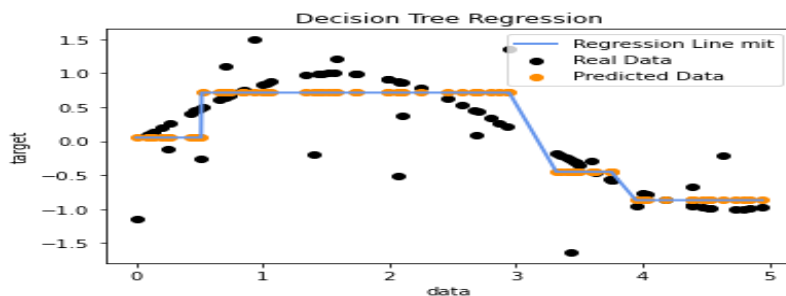
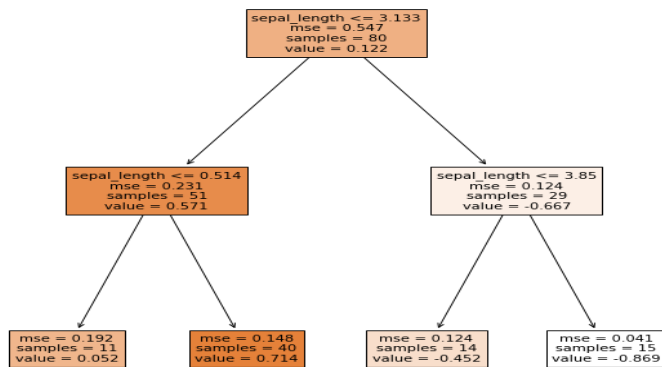
- Viele Hyperparameter, deshalb Tuning notwendig.
- Längere Trainingszeit bei großen Datenmengen.
- Erklärbarkeit Ergebnisse schwierig (Hyperebene n-dimensionaler Raum).

Klassifikation: Algorithmus bestimmt „Hyperebene“¹, die Datensatz am besten in die gewünschte Anzahl Klassen einteilt. Dabei wird die Hyperebene so gelegt, daß die Grenze zwischen den Klassen möglichst breit ist.

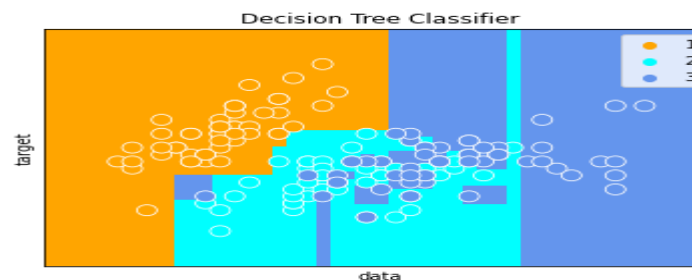
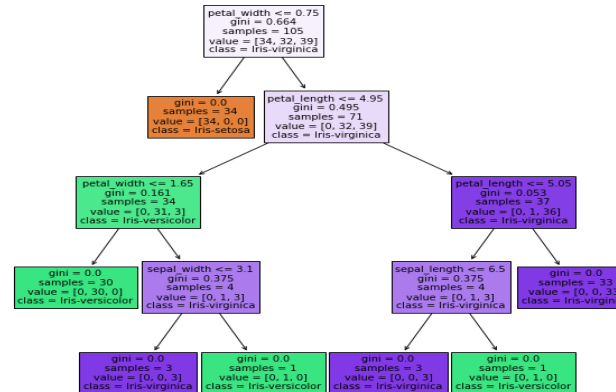
Regression: Algorithmus bestimmt „Hyperebene“ und nimmt Punkt, der möglichst nah an der Hyperebene liegt

SUPERVISED LEARNING: DECISION TREE.

Regression kontinuierliche Variable



Klassifizierung diskrete Variable



Vorteile:

- Einfach verständliche Ergebnisse.
- Einsetzbar für Regression und Klassifikation.
- Weniger Aufwand Datenaufbereitung (keine Skalierung/ Normierung).
- Robust bei fehlenden Daten.
- Robuster auch bei vieldimensionalen Features.

Nachteile:

- Nicht stabil gegenüber Updates (dann muß erneut trainiert werden).
- Schlechte Übertragbarkeit auf ähnliche Daten.
- Höherer Zeit-/ Ressourcenbedarf für Training.

Klassifikation: wiederholtes Aufteilen Datenset in zwei Untermengen anhand des Features mit höchstem Informationsgehalt.
Regression: erfolgt durch Bestimmen des Durchschnitts über die y-Werte des finalen Endknoten.

4. ENSEMBLE LEARNING

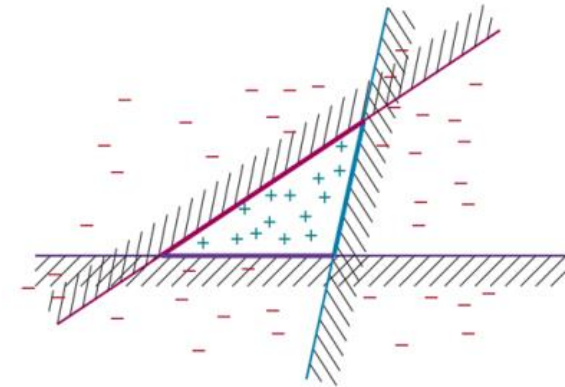
SUPERVISED LEARNING: ENSEMBLE LEARNING

Ziel: Verbesserte Performance durch die Kombination einzelner Lernverfahren (und Ausgleichen deren Schwächen).

Ermöglicht:

- Höhere Genauigkeit (Reduktion Streuung und Verzerrung).
- Mehr Anwendungsfelder können abgedeckt werden.

Aber: Höherer Ressourcenbedarf für Training.

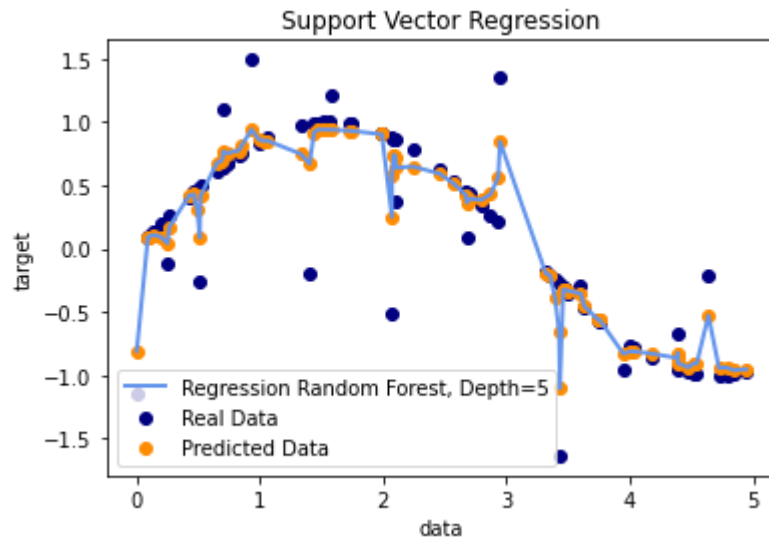


Häufigst eingesetzte Verfahren:

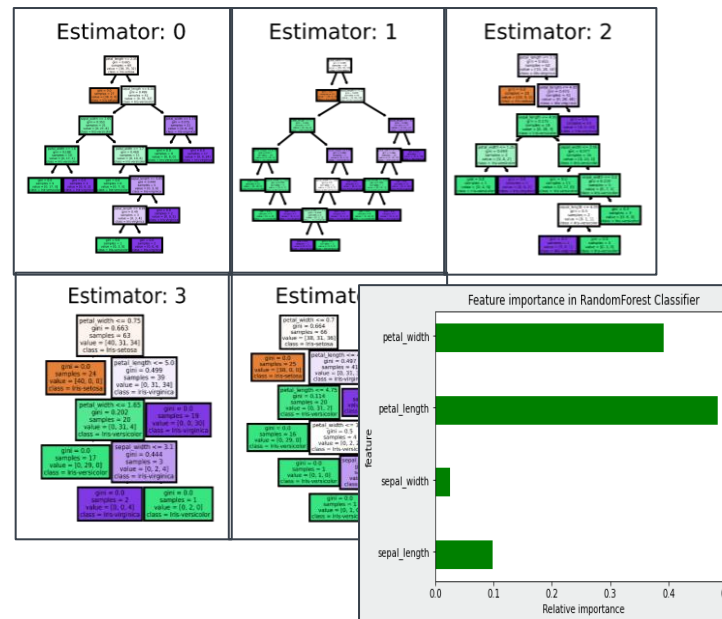
- Bagging: Kombination verschiedener Lernverfahren und **gleiche** Gewichtung (bspw. Durchschnitt) → Random Forest.
- Boosting: Kombination "schwacher" Lernverfahren per **performance-abhängiger Gewichtung** → XGBoost.

SUPERVISED LEARNING: RANDOM FOREST.

Regression kontinuierliche Variable



Für diskrete, mehrere Klassen



Vorteile:

- Parallelisierbarkeit Training.
- Feature Importance: Algorithmus liefert Aussage, was die wichtigsten Features für die Vorhersage sind.
- Einsetzbar für hochdimensionale Daten.
- Höhere Genauigkeit als Entscheidungsbäume (Reduziert Overfitting durch Gewichtung mehrerer Entscheidungsb.).

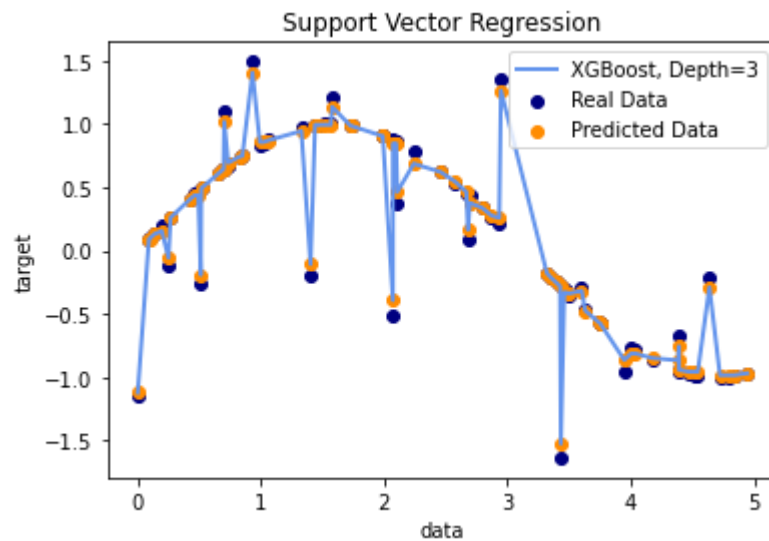
Nachteile:

- Komplexer als Entscheidungsbäume.
- Nicht mehr so leicht interpretierbar.
- Schlechtere Performance bei Regression.
- Schlechtere Laufzeit-Performance.

Einsatz verschiedener Entscheidungsbäume und danach Kombination der Entscheidungsbäume. Dabei wird jeder Entscheidungsbaum gleich gewichtet, d. h. hat gleichen Einfluß auf das Ergebnis.

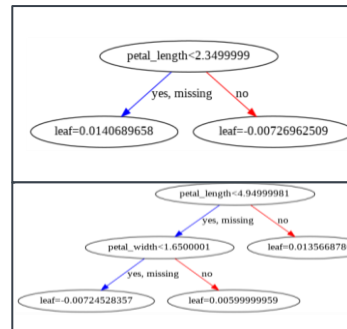
SUPERVISED LEARNING: XGBOOST.

Regression kontinuierliche Variable

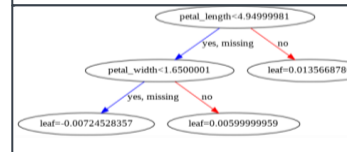


Für diskrete, mehrere Klassen

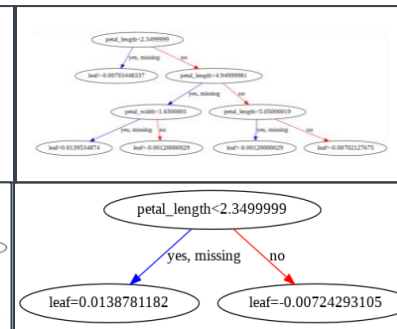
Num_Trees=0



Num_Trees=2



Num_Trees=1



Num_Trees=3

Vorteile:

- Hervorragende Performance für strukturierte Daten.
- Sehr hohe Genauigkeit.
- Parallelisierbarkeit Training.
- Feature Importance (was sind die wichtigsten Features für Vorhersage?).

Nachteile:

- Erklärbarkeit Algorithmus.
- Arbeitet nur mit numerischen Werten (Kategorische Variablen müssen in 1-dimensionale Features je Wert umgewandelt werden).
- Hyperparameter-Tuning notwendig für Vermeiden Overfitting.

Einsatz mehrerer Modelle mit dem Ziel, durch die Hinzunahme zusätzlicher Modelle die Fehlklassifikationen der bisherigen Modelle zu reduzieren (Start mit Modell 1, dann trainiere Modell 2 auf Fehler von Modell 1, ...).



5. VERGLEICH/ GEGENÜBERSTELLUNG VERFAHREN

GEGENÜBERSTELLUNG ERGEBNISSE ALGORITHMEN AM BEISPIEL VON DREI DATENSÄTZEN.

Kein Tuning Parameter eingesetzt,
Source Code liegt im Github.

ALGORITHMUS	REGRESSION MODIFIZIERTE SINUS-KURVE	IRIS CLASSIFICATION AUF 2 SEPAL-FEATURES (SEHR KLEINE DATENMENGE)	IRIS CLASSIFICATION AUF 4 FEATURES (KLEINE DATENMENGE)	TITANIC CLASSIFICATION (KLEINE DATENMENGE)
Lineare Regression	0.43	0.82	0.98	0.80
Naive Bayes	0.43	0.78	1.0	0.79
Entscheidungsbaum	0.76	0.93	0.98	0.79
K-Nearest Neighbours	0.80	0.83	0.98	0.81
Support Vector Machine/ Regressor	0.77	0.82	0.98	0.79
Random Forest	0.91	0.93	0.98	0.82
XGBoost	0.99	0.82	0.98	0.79

Nächste Woche(n):
Verfahren für große/ sehr
große Datenmengen

Man weiß sehr häufig nicht im Voraus, welche Algorithmen am besten für einen Datensatz performen.
Deshalb Empfehlung: verschiedene Algorithmen ausprobieren und deren Parameter tunen



6. CASE STUDY

CASE STUDY IN GRUPPENARBEIT

1. Erstellen Sie ein Notebook in Google Colab.
2. Laden Sie die Standard-Bibliotheken (vgl. Notebooks „VL 6 Supervised Learning-Update“ und „Case Study supervised“).
3. Laden Sie den Datensatz per Pandas-Funktion `read_csv()`.
4. Wenden Sie den gelernten Workflow des Supervised learnings auf den Datensatz an.
 1. Probieren Sie verschiedene Verteilungen Trainings- und Testmengen aus. Wie ändern sich die Ergebnisse?
 2. Wählen Sie 2 verschiedene „simple“ Algorithmen des überwachten Lernens an.
 - Welcher Algorithmus liefert die höchste Genauigkeit?
 - Wie ändern sich die Ergebnisse, falls Sie die Hyperparameter des Algorithmus ändern?
 3. Wenden Sie XGBoost und RandomForest an.
 - Gibt es Unterschiede bzgl. Genauigkeit?
 - Wie sind die Unterschiede zu den „simplen“ Algorithmen?
5. Stellen Sie Ihre Ergebnisse, Hypothesen und Plots vor.



ZUSAMMENFASSUNG

ZUSAMMENFASSUNG SUPERVISED LEARNING.

Vorteile:

- Wir erhalten „bewertete“ Ergebnisse.
- Eigenständiges Handeln des Algorithmus möglich.

Nachteile:

- Bestimmen Zielvariable (Labeln) wichtig und aufwendig.
- Datenqualität sehr wichtig.
- Kein Algorithmus perfekt, Tuning notwendig (v.a. Hyperparameter).

LITERATUR UND WEITERE QUELLEN (AUSZUG).

Künstliche Intelligenz:

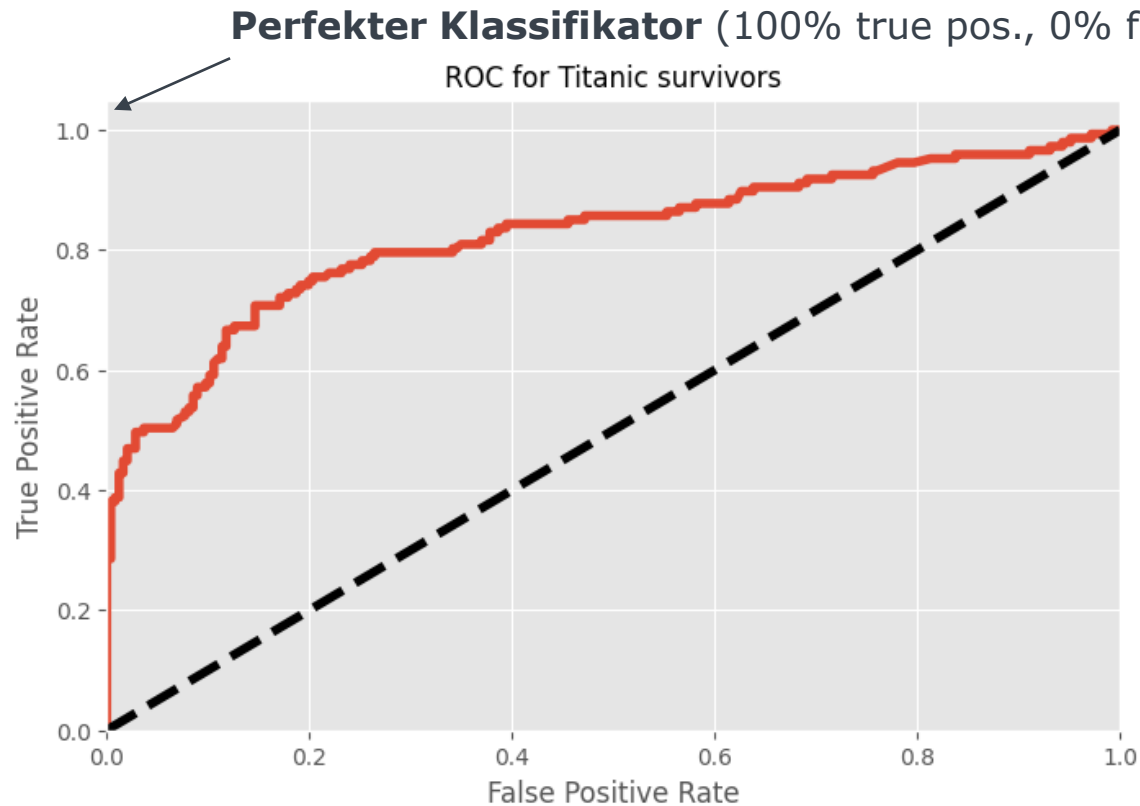
- Gröner, Heinecke: Kollege KI
- Burkov: The Hundred-Page Machine Learning Book, online verfügbar unter [Link](#)
- Nielsen: Neural Networks and Deep Learning, online verfügbar unter [Link](#)
- Russel, Norvig: Artificial Intelligence – a modern approach
- Produktentwicklung mit AI:
 - Ameisen: Building Machine Learning Powered Applications: Going from Idea to Product
 - Ng: Machine Learning Yearning, online verfügbar unter [Link](#)

Kostenfreie Online-Kurse (bei Interesse):

- Python-Kurse
 - Python for Everybody ([Link](#))
 - Udacity Python Course ([Link](#))
 - **Coursera Course Deep Learning** ([Link](#))
 - FAST AI ([Link](#))

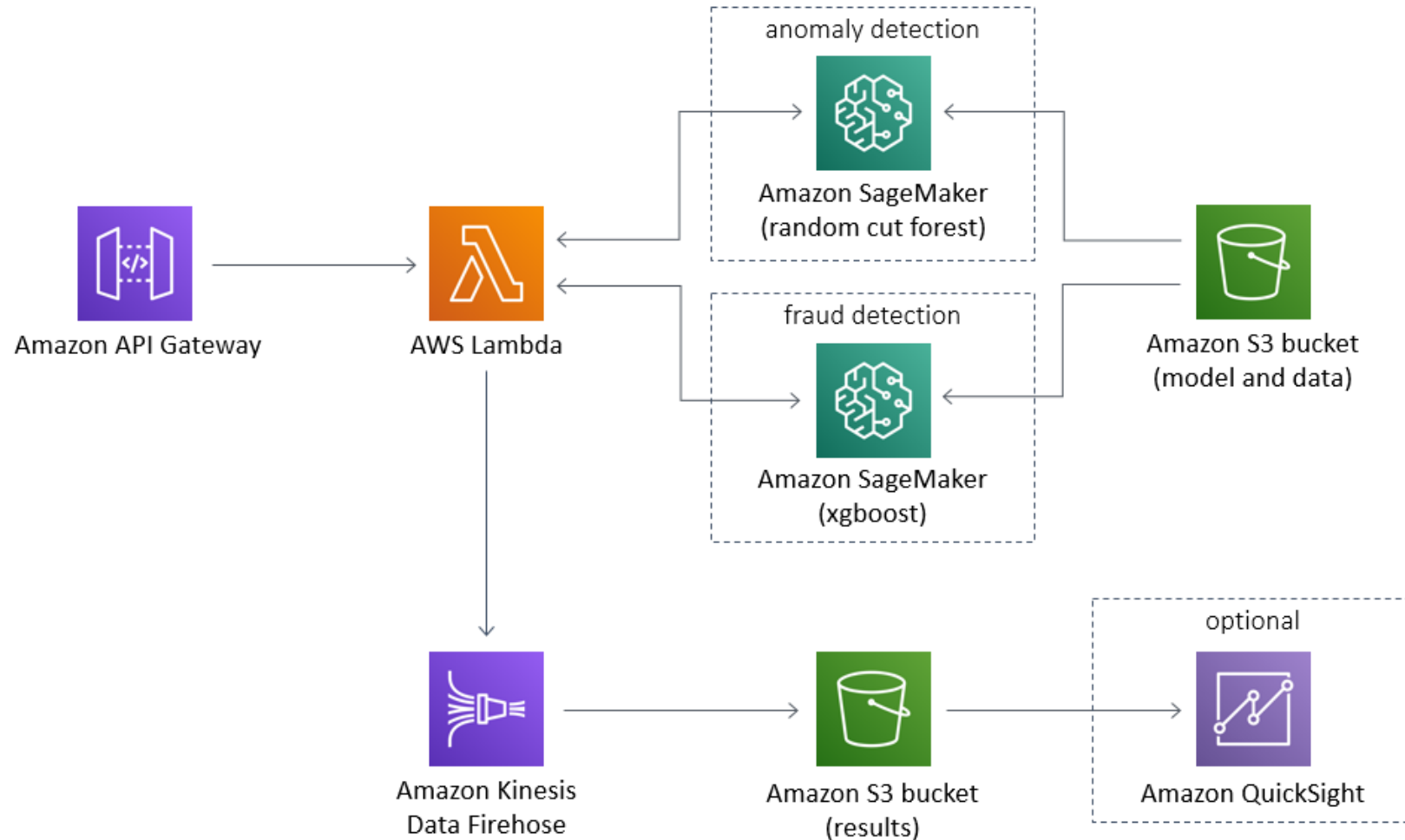
BACKUP

DETAILLIERUNG KLASSIFIZIERUNGSMETRIKEN. ROC/ AUC.



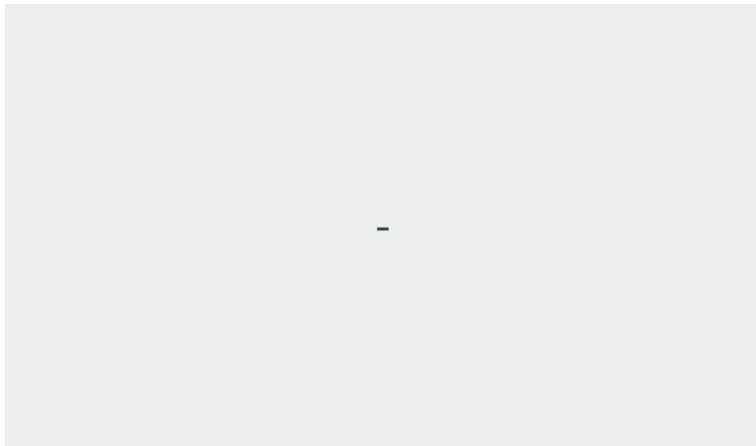
- **ROC**(Receiver-Operating-Characteristics): Darstellung prozentuales Verhältnis zwischen den richtig und falsch als wahr prädizierten Werten.
- **AUC**(Area under ROC Curve): Integral der ROC-Kurve. Je höher AUC-Fläche, desto besser der Klassifikator.

INTEGRATION MACHINE LEARNING IN PRAXIS AM BEISPIEL FRAUD DETECTION.

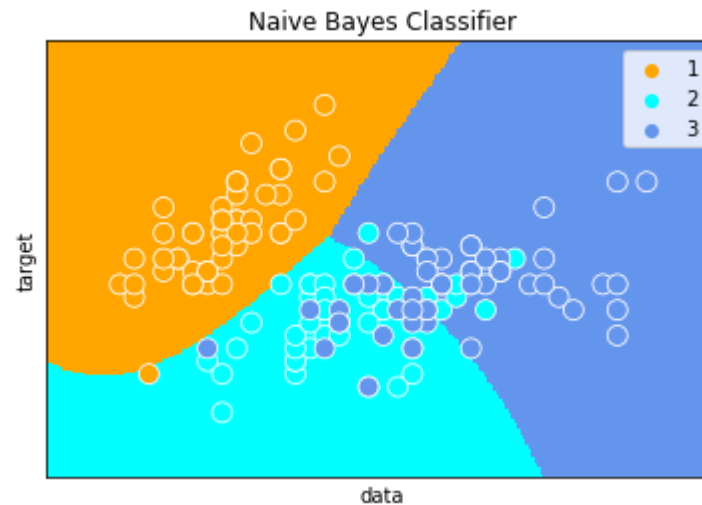


SUPERVISED LEARNING: NAIVE BAYES.

Regression kontinuierliche Variable



Klassifizierung diskrete Variable



Vorteile:

- Kann gut mit hochdimensionalen Daten umgehen.
- Wenig Trainingsdaten notwendig.
- Schnell berechenbarer Algorithmus.
- Oft bessere Genauigkeit als kompliziertere Modelle (aber nur falls Variablen statistisch unabhängig).

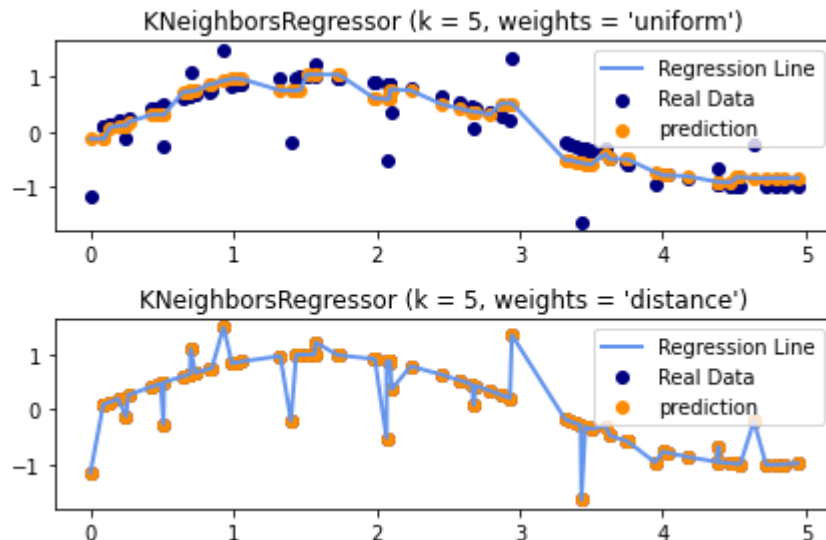
Nachteile:

- Annahme bedingte Unabhängigkeit nicht immer gegeben.
- Bei großen Datenmengen liefern andere Algorithmen bessere Ergebnisqualität (wegen Bayes-Regel für Updates).

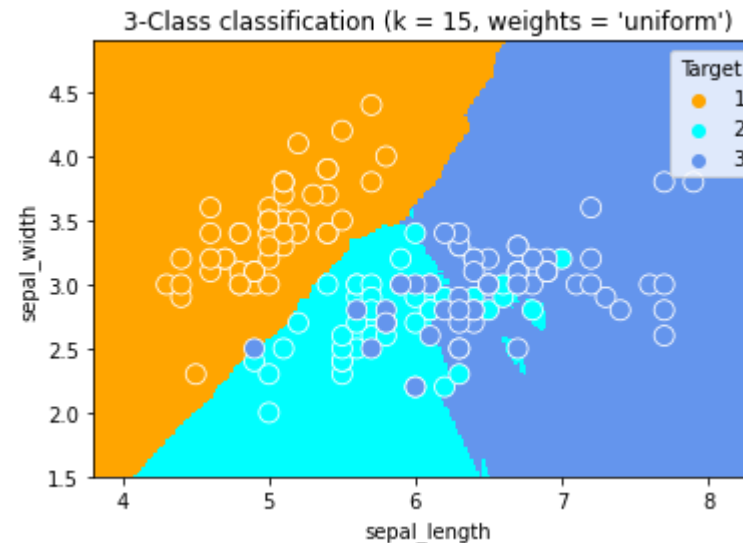
Nutzt Bayes-Theorem für das Berechnen der Zugehörigkeit eines Features zu einer Klasse und ordnet das Feature der Klasse zu, für die die berechnete Wahrscheinlichkeit am höchsten ist.

SUPERVISED LEARNING: K NEAREST NEIGHBOURS.

Regression kontinuierliche Variable



Klassifizierung diskrete Variable



Vorteile:

- Einfach verständlich.
- Einsetzbar für Regression und Klassifikation.
- Ohne Labels und ohne Training einsetzbar.
- Nur 1 Hyperparameter.

Nachteile:

- Lernt nicht aus Trainingsdaten.
- Hoher Speicherbedarf (ganzer Datensatz wird in Speicher abgelegt).
- Ausreißer/ fehlende Daten haben großen Einfluß auf Algorithmus.
- Probleme mit hochdim. Daten.
- Selten eingesetzt für Regression.

Klassifikation eines Punktes erfolgt anhand Mehrheitsvotum seiner k-Nachbarn.
Regression erfolgt anhand Bilden des gewichteten Durchschnitt der Zielwerte der k-Nachbarn.