Sitzung 2 – Version 1

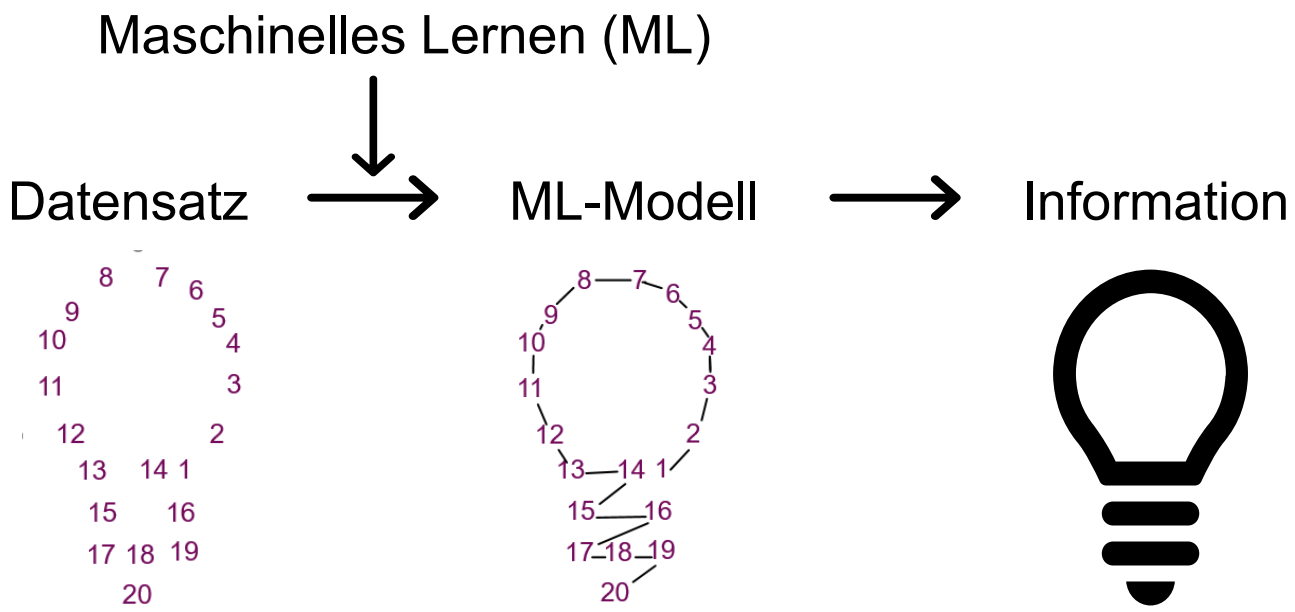
Inhalt

1 Grundbegriffe des Maschinellen Lernens

Maschinelles Lernen: Überblick

„Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience.“

(Tom Mitchell, 1997)



Arten des Maschinellen Lernens

- Überwachtes Lernen (Supervised Learning):
  Das Verfahren untersucht einen Datensatz bezüglich einer expliziten Fragestellung. Hierbei soll am Ende ein Merkmal vorhergesagt werden, das so genannte Label.

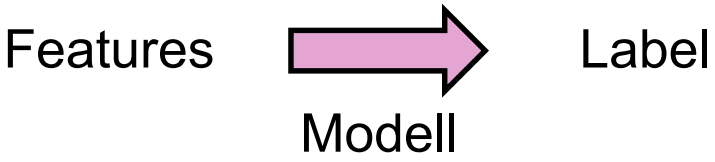
- Unüberwachtes Lernen (Unsupervised Learning):
  Das Verfahren untersucht einen Datensatz auf vorher unbekannte Muster.

- Bestärkendes Lernen (Reinforcement Learning):
  Das Verfahren entwickelt durch Ausprobieren und (positiver oder negativer) Bestärkung eine Strategie in einem gegebenen Problem.

Überwachtes Lernen: Vorgehen

- Datensatz:
  Unter den Merkmalen (Features) existiert ein Zielmerkmal (Label).

- Ziel:
  Entwicklung einer Zuordnung von Features zum Label:



- Vorgehen:
  Systematische Untersuchung des Datensatzes, um daraus eine geeignete Zuordnung zu entwickeln.

2 Das Stahlprojekt

2.1 Das Problem und die Daten

2.2 Ideen zur Klassifikation

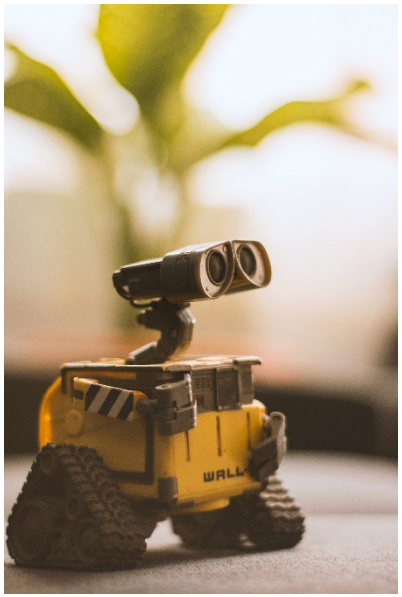


Foto von Lenin Estrada von Pexels

Lernziele

1. Sie benennen unterschiedliche Ideen, wie ein Modell zur Klassifizierung von Datensätzen entstehen kann.

2. Sie erkennen die Visualisierung mittels Streudiagramm als wichtiges Hilfsmittel, um ein Modell zur Klassifizierung von Datensätzen zu entwickeln und darzustellen.

2.1 Das Problem und die Daten

Stahlprojekt: Ausgangssituation

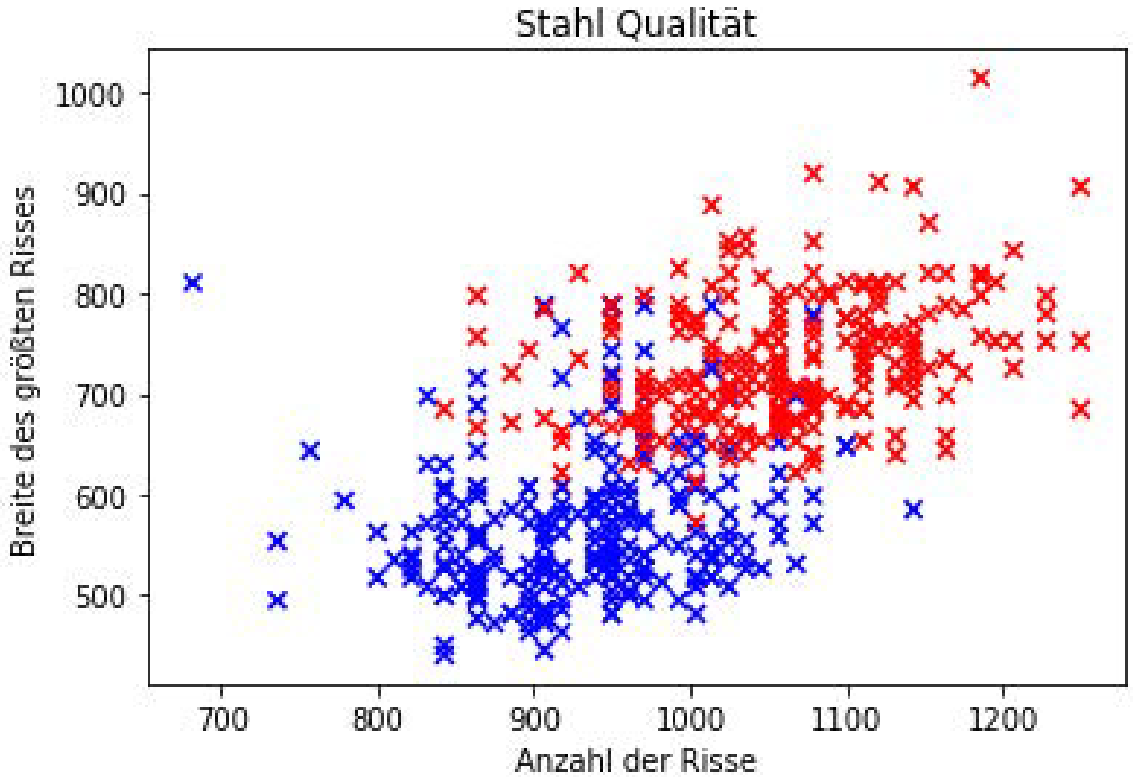
Sie arbeiten für eine Firma, die Bauteile aus Stahl herstellt und vertreibt.

Die Entwicklungsabteilung Ihrer Firma hat einen neuen Stahl ausgewählt, welcher unter hoher Belastung besonders langlebig sein soll. In einem bestehenden Produktionsprozess für Bauteile könnte dieser Stahl den bisher verwendeten Stahl ersetzen.

Es soll nun ein ML-Modell entwickelt werden, um aus der Anzahl der Risse & der Breite des größten Risses die Stahlqualität zu beurteilen.

Foto von Martinelle auf Pixabay

Stahlprojekt: Zugrundliegender Datensatz



2.2 Ideen zur Klassifikation

B

Stahlprojekt: Entwicklung erster Modelle

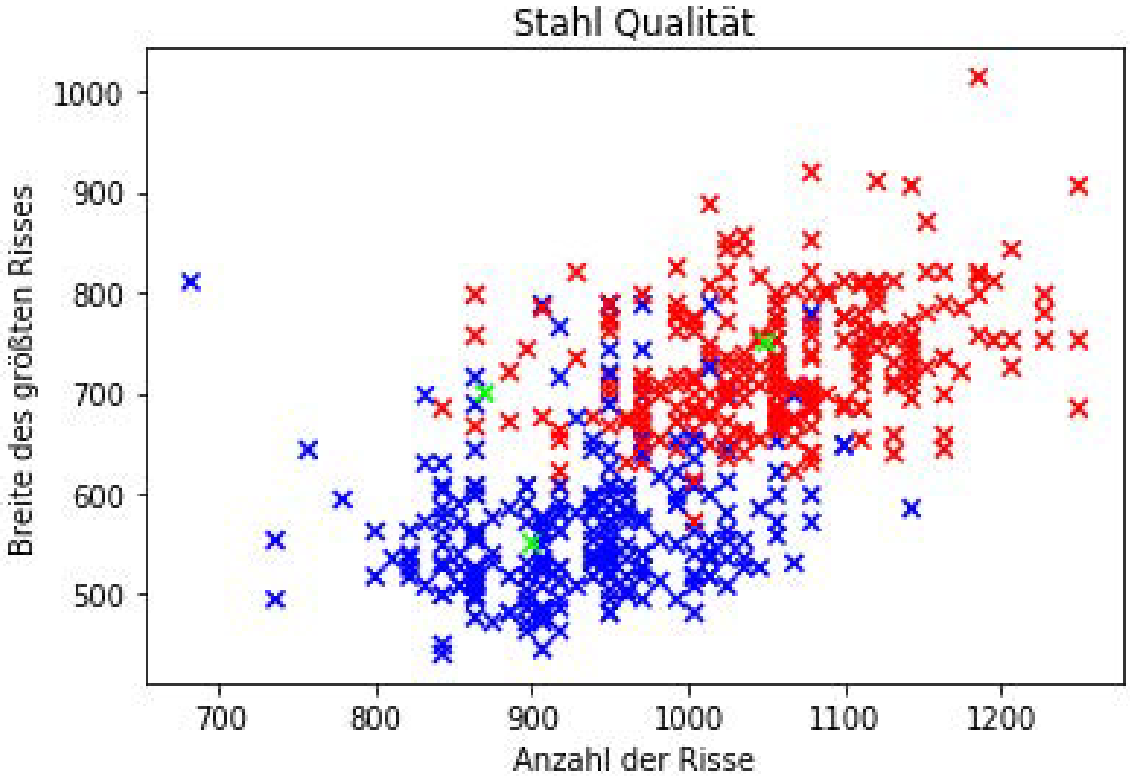
Auf der folgenden Folie sind in grün drei neue Beispiele im bekannten Datensatz markiert.

Arbeitsauftrag:

Welcher Klasse würden Sie diese Beispiele zuordnen und warum? Formulieren Sie mindestens zwei unterschiedliche Modelle, also eine Zuordnung von Features zum Label, zur Klassifikation neuer Beispiele. Visualisieren Sie Ihre Modelle außerdem in der abgebildeten Grafik. Nutzen Sie hierfür Mural (Link folgt in Veranstaltung).

Hilfestellung: Formulieren Sie die Zuordnungen in der Form „Wenn für das neue Beispiel (…) gilt, dann wird das Beispiel der Klasse (…) zugeordnet.“

Stahlprojekt: Entwicklung erster Modelle



LÖSUNG

Stahlprojekt: Entwicklung erster Modelle

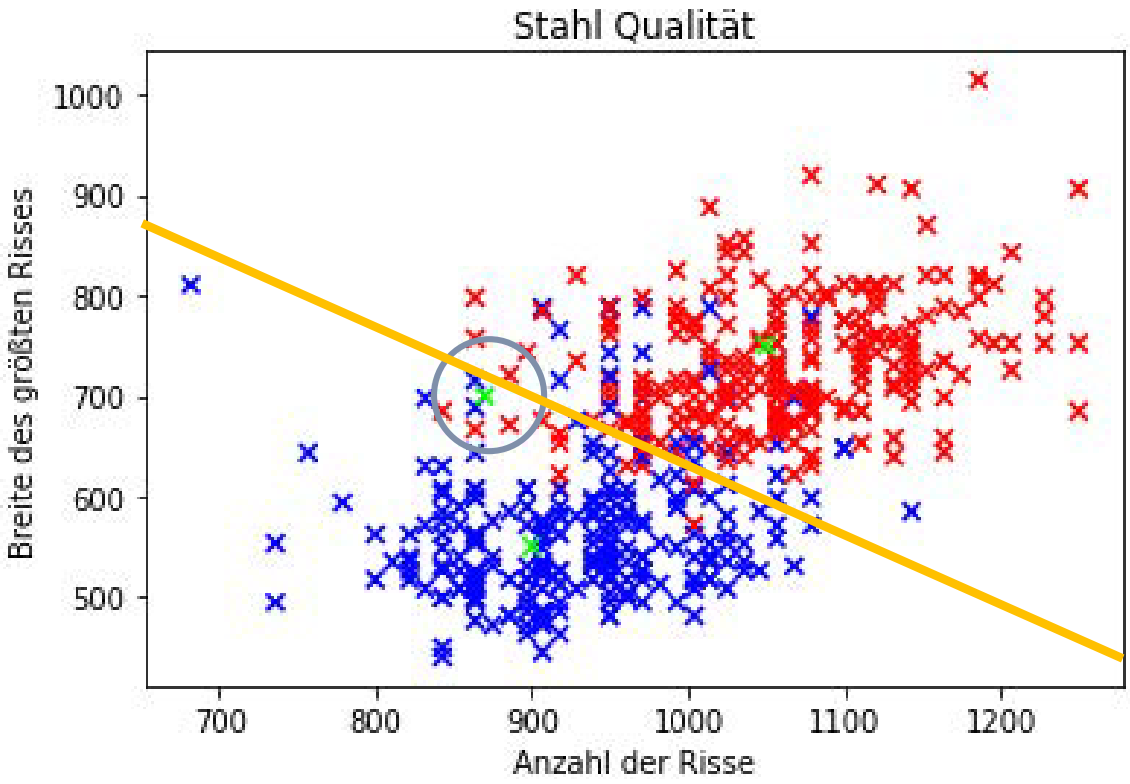
Möglichkeiten:

- Wenn für eine festgelegte Anzahl naheliegender anderer Beispiele eine Mehrheit bezüglich einer Klasse besteht, wird das neue Beispiel dieser Klasse zugeordnet.

- Es wird eine Linie durch die Punktewolke gelegt. Wenn ein Beispiel unterhalb der Linie liegt, gehört es zu einer Klasse, wenn nicht, gehört es zur anderen Klasse.

LÖSUNG

Stahlprojekt: Entwicklung erster Modelle



Stahlprojekt: Entwicklung erster Modelle

Um die Begrifflichkeiten im Kontext ML zu wiederholen, sollen die Modelle nun noch einmal neu formuliert werden:

Arbeitsauftrag zur Nachbereitung der Veranstaltung:
Formulieren Sie Ihre Modelle so um, dass Sie die Begriffe *Label*, *Feature* und *Ausprägung* mindestens einmal benutzen.

LÖSUNG

Stahlprojekt: Entwicklung von ersten Modellen

Möglichkeiten:

- Es werden die Kreuze betrachtet, die in einem bestimmten Umkreis zu dem Kreuz aus der Kombination der Featureausprägung des neuen Beispiels liegen. Das Label des neuen Beispiels entspricht dann der Klasse der meisten umgebenen Kreuze.

- Es wird eine Gerade durch die Punktwolke gelegt. Das Label des neuen Beispiels hängt dann davon ab, ob das Kreuz aus der Kombination der Featureausprägung unter oder über der Geraden liegt.

3 k Nearest Neighbor-Klassifikation

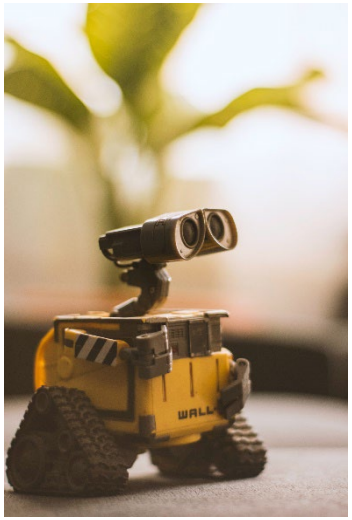


Foto von Lenin Estrada von Pexels

Lernziele

1. Sie erläutern die Idee der kNN-Klassifikation unter Zuhilfenahme einer passenden Visualisierung.

2. Sie erkennen den Zusammenhang zwischen dem umgangssprachlichen Abstand und der mathematischen Metrik und benennen Beispiele für unterschiedliche Metriken.

3. Sie folgern die Funktionsweise längerer Codeabschnitte unter Verwendung der zugehörigen Outputs und unterscheiden verschiedene Datentypen in Python.

4. Sie erläutern das Vorgehen der kNN-Klassifikation unter Zuhilfenahme des Pseudocodes.

Lernziele

5. Sie bereinigen Datensätze und bereiten sie durch Trennung in Trainingsdaten und Testdaten für die Anwendung der kNN-Klassifikation mit scikit-learn in Python vor.

6. Sie fitten zu den Trainingsdaten ein Modell nach der Idee der kNN-Klassifikation mit scikit-learn in Python.

7. Sie visualisieren mit scikit-learn gefittete Modelle in Python.

8. Sie erkennen die Metrik d und den Faktor k als Stellschrauben, welche das Ergebnis des mit der kNN-Klassifikation entstehenden Modells beeinflussen.

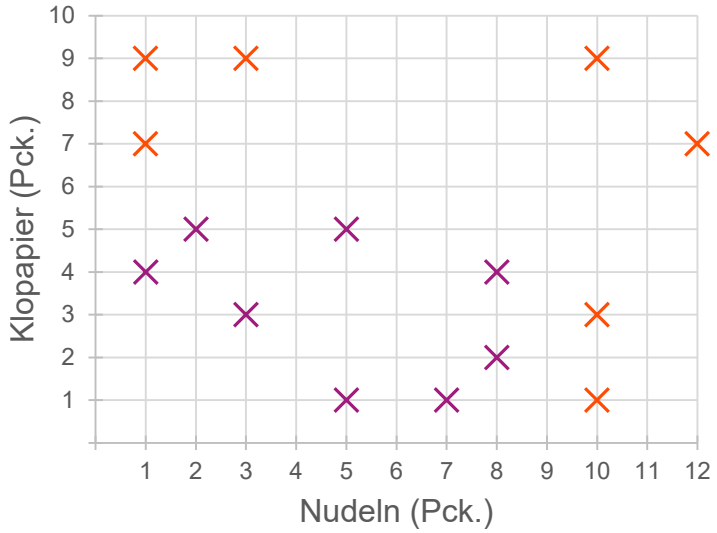
3.1 Idee der k Nearest Neighbor-Klassifikation

Konstruiertes Beispiel: Einkaufswagen-Klassifikation

Ausgangsanliegen: Automatisierte Unterscheidung zwischen Hamsterkauf und Familieneinkauf.

Datengrundlage:

Nudeln (Pck.)	Klopapier (Pck.)	Klasse
1	9	Hamsterkauf
1	7	Hamsterkauf
3	9	Hamsterkauf
10	1	Hamsterkauf
10	3	Hamsterkauf
12	7	Hamsterkauf
10	9	Hamsterkauf
1	4	Familieneinkauf
2	5	Familieneinkauf
3	3	Familieneinkauf
5	1	Familieneinkauf
5	5	Familieneinkauf
7	1	Familieneinkauf
8	2	Familieneinkauf
8	4	Familieneinkauf



B

Konstruiertes Beispiel: Einkaufswagen-Klassifikation

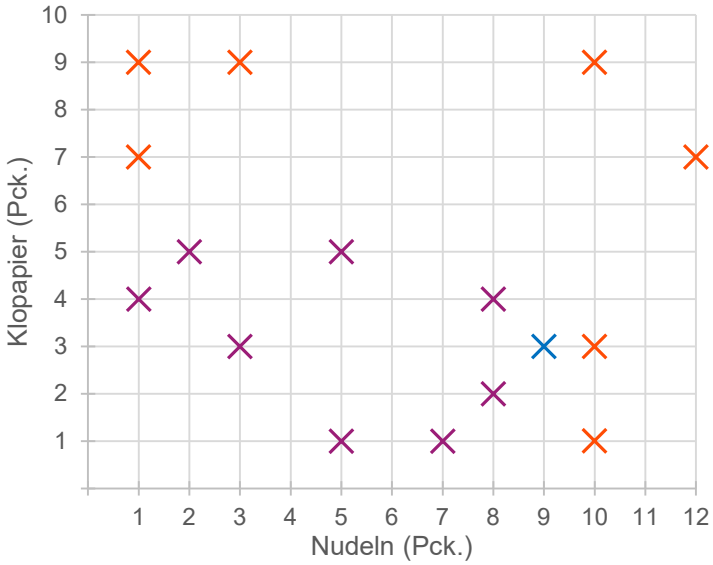
In der Grafik ab Folie 25 ist ein neues Beispiel (ein nicht klassifizierter Einkaufswagen, markiert durch das blaue Kreuz) hinzugefügt worden. Auf den nachfolgenden Folien ist die k Nearest Neighbor-Klassifikation (auch: kNN-Klassifikation) für einen, drei und fünf Nachbarn dargestellt.

Arbeitsauftrag:

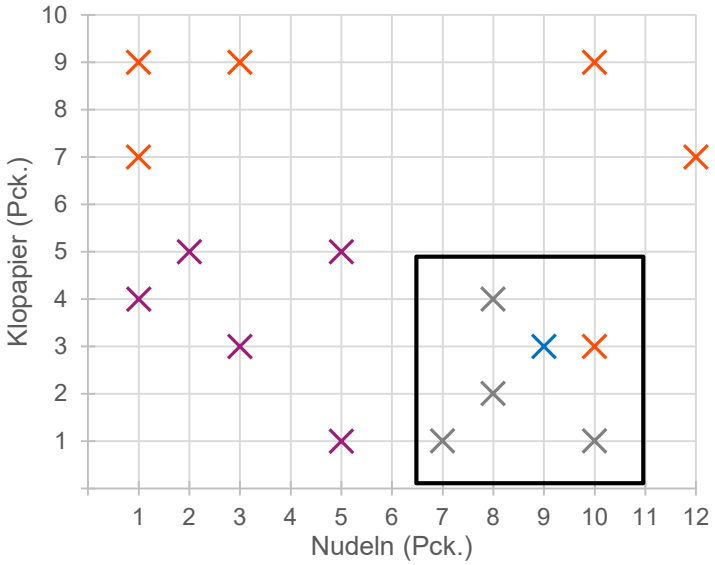
Diskutieren und notieren Sie die Antworten auf die folgenden Fragen:

- Nach welcher Zuordnung wird hier klassifiziert?
- Was bedeutet der Begriff „Nachbarn“ im Kontext der Klassifikation?
- Welche Schwierigkeiten fallen Ihnen zu diesem Vorgehen ein?

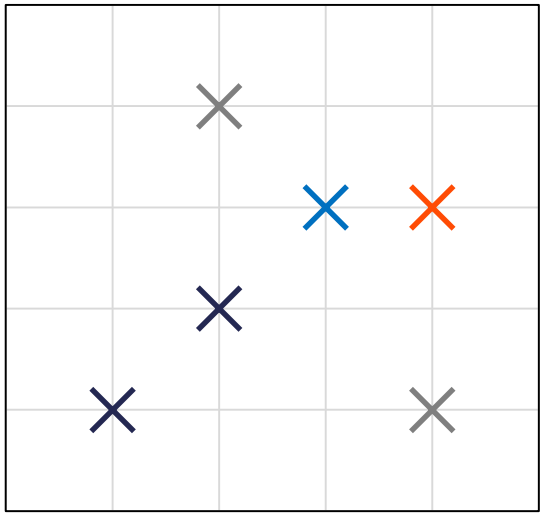
Konstruiertes Beispiel: Einkaufswagen-Klassifikation



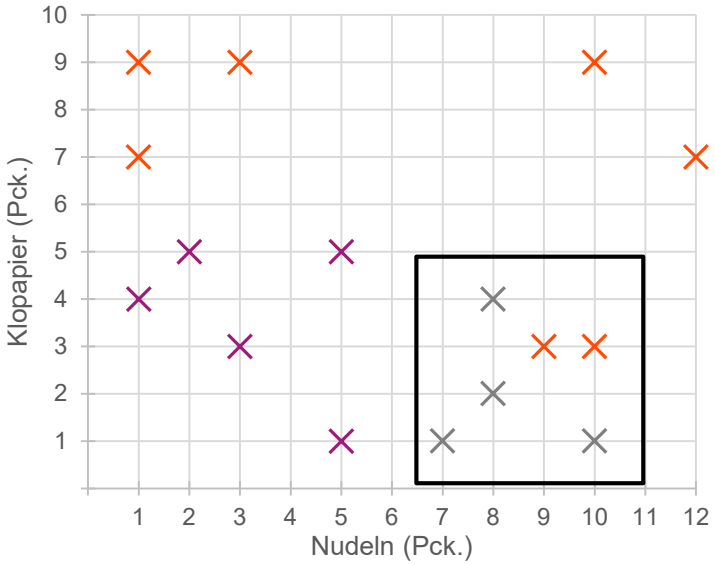
Konstruiertes Beispiel: Einkaufswagen-Klassifikation



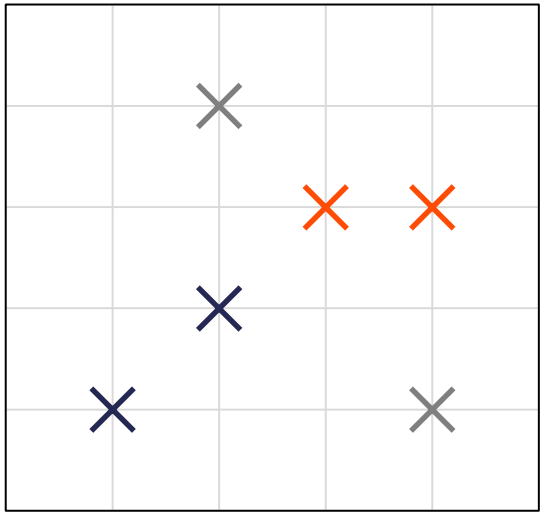
Ausgangssituation $k = 1$



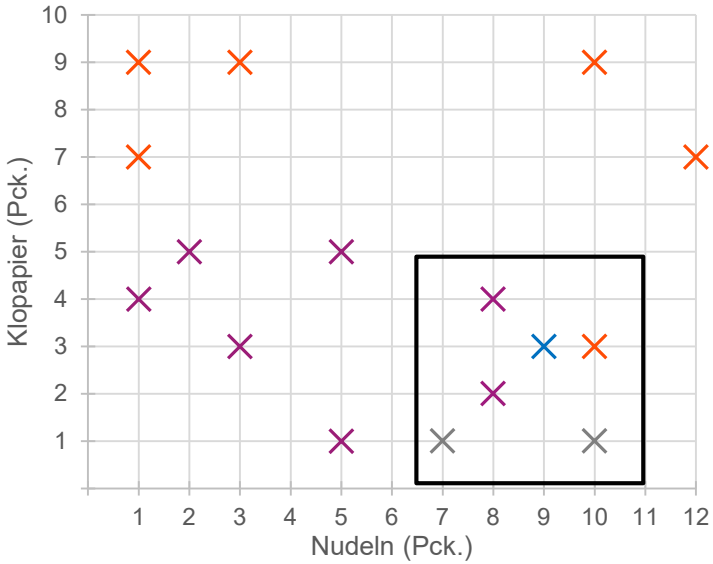
Konstruiertes Beispiel: Einkaufswagen-Klassifikation



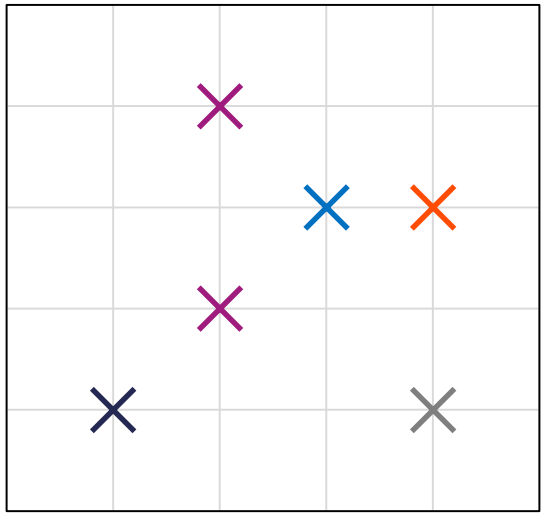
Klassifikation $k = 1$



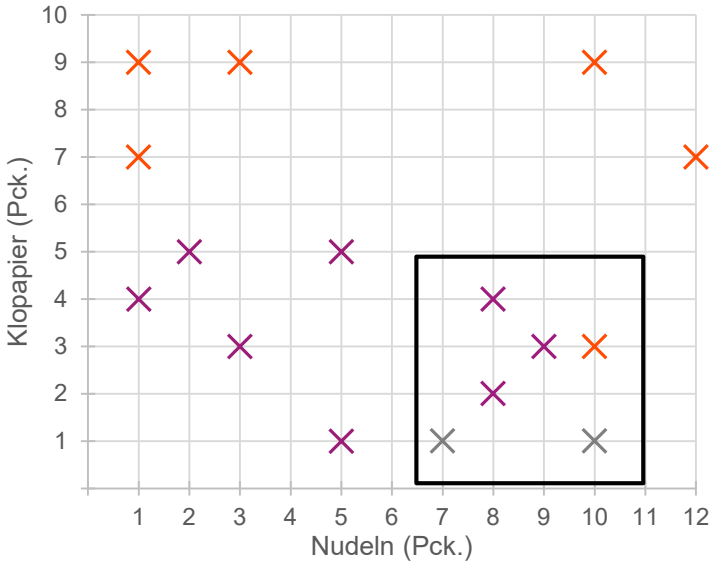
Konstruiertes Beispiel: Einkaufswagen-Klassifikation



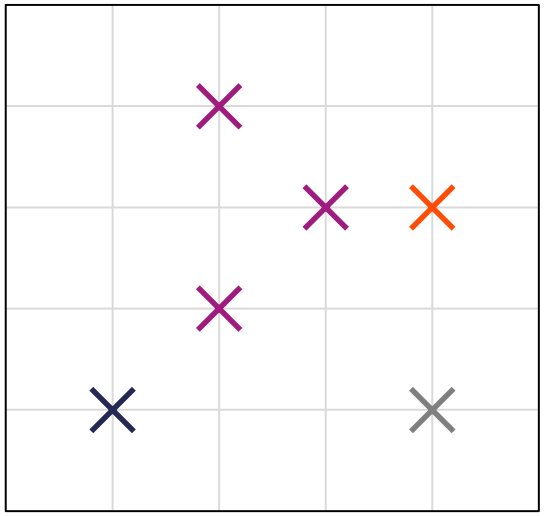
Ausgangssituation $k = 3$



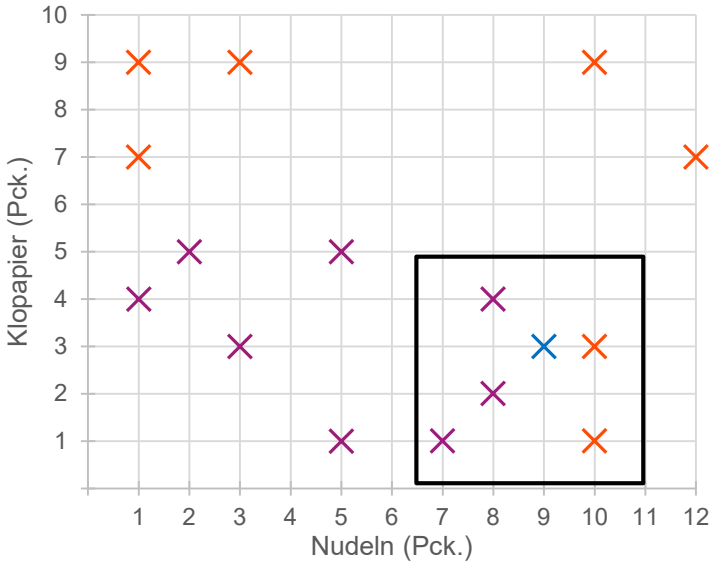
Konstruiertes Beispiel: Einkaufswagen-Klassifikation



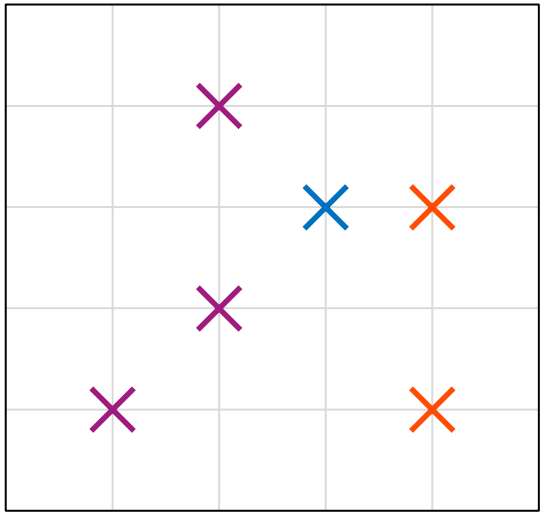
Klassifikation $k = 3$



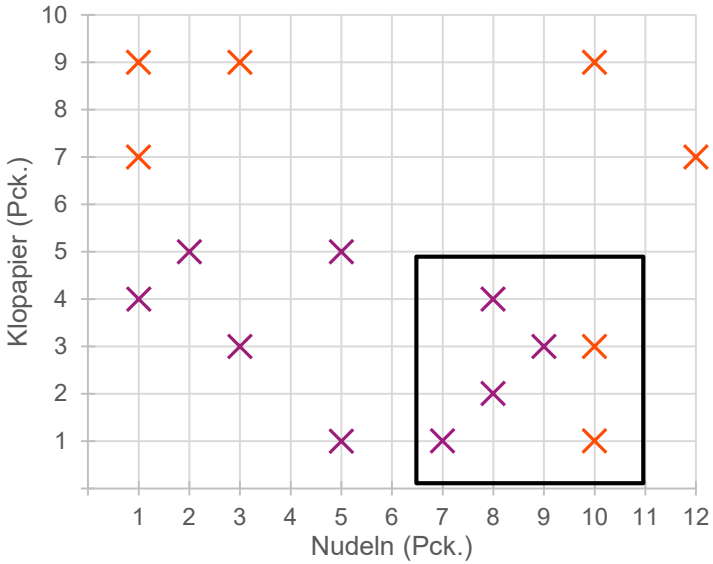
Konstruiertes Beispiel: Einkaufswagen-Klassifikation



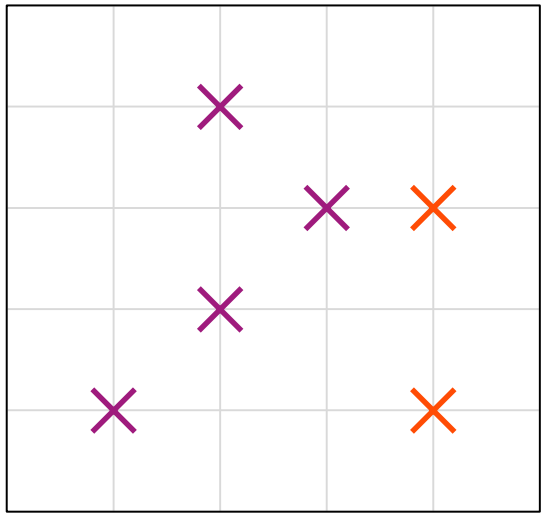
Ausgangssituation $k = 5$



Konstruiertes Beispiel: Einkaufswagen-Klassifikation



Klassifikation $k = 5$



LÖSUNG

Fragen zur Einkaufswagen-Klassifikation

- Die Klasse entspricht der am häufigsten vorliegenden Klasse unter einer vorgegebenen Anzahl umliegender Beispiele.
- Als „Nachbarn“ gelten die Beispiele, die in der Grafik nahe an dem zu klassifizierenden Beispiel liegen.
- Gleichheit zwischen den Klassen (z.B. wenn gleich viele Nachbarn pro Klasse oder zwei Nachbarn gleich weit entfernt)

k Nearest Neighbor-Klassifikation: Zusammenfassung erster Ergebnisse

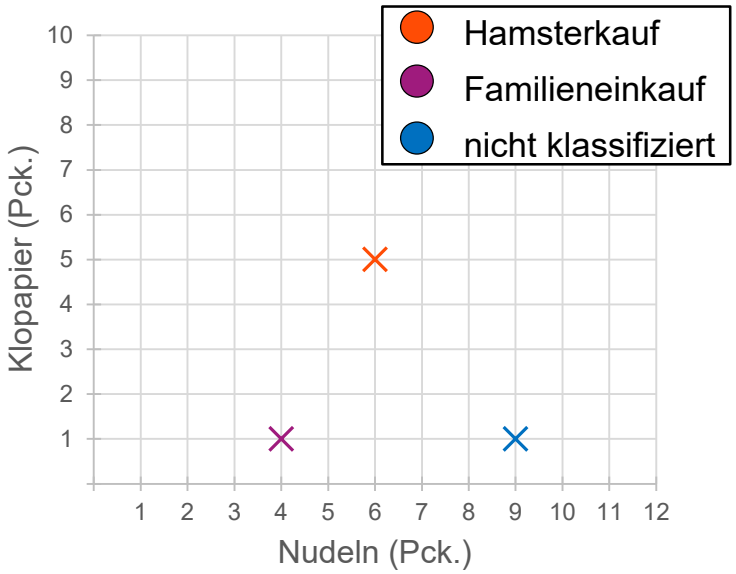
Wird die k Nearest Neighbor-Klassifikation anhand von $k \geq 1$ nächsten Nachbarn vorgenommen, ist das Vorgehen zur Klassifikation eines neuen Beispiels wie folgt:

- Bestimmung der k nächsten Nachbarn des neuen Beispiels.
- Bestimmung der am häufigsten vertretenen Klasse unter den k Nachbarn.
- Klassifizierung des neuen Beispiels in diese Klasse.

3.2 Formalisierung der k Nearest Neighbor-Klassifikation

Konstruiertes Beispiel: Einkaufswagen-Klassifikation II

In der nebenstehenden Grafik sind ein Hamstereinkauf $((6/5))$, ein Familieneinkauf $((4/1))$ und ein nicht klassifizierter Einkaufswagen $((9/1))$ dargestellt.



Arbeitsauftrag:

Versuchen Sie mit der k Nearest Neighbor-Klassifikation für $k = 1$ zu begründen, zu welcher Klasse der neue Einkaufswagen klassifiziert werden sollte. Versuchen Sie dabei sowohl einen Grund für den Hamsterkauf als auch einen Grund für den Familieneinkauf zu finden.

Unterschiedliche Metriken in $\mathbb{R}^2$

Es werden Punkte $a = (x_a, y_a)$ und $b = (x_b, y_b)$ im $\mathbb{R}^2$ betrachtet. Es gibt unterschiedliche Möglichkeiten, den Abstand $d(a, b)$ der Punkte zu bestimmen, beispielsweise:

- Euklidische Metrik:

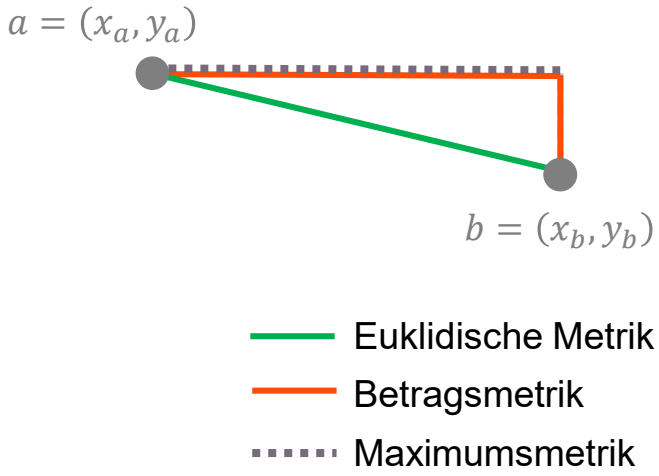
  $$d(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

- Betragsmetrik:

  $$d(a,b) = |x_a - x_b| + |y_a - y_b|$$

- Maximumsmetrik:

  $$d(a,b) = \max\{|x_a - x_b|, |y_a - y_b|\}$$



Konstruiertes Beispiel: Einkaufswagen-Klassifikation II

Die drei Wege werden nun genutzt, um die Abstände im Beispiel zu bestimmen:

- Euklidische Metrik:

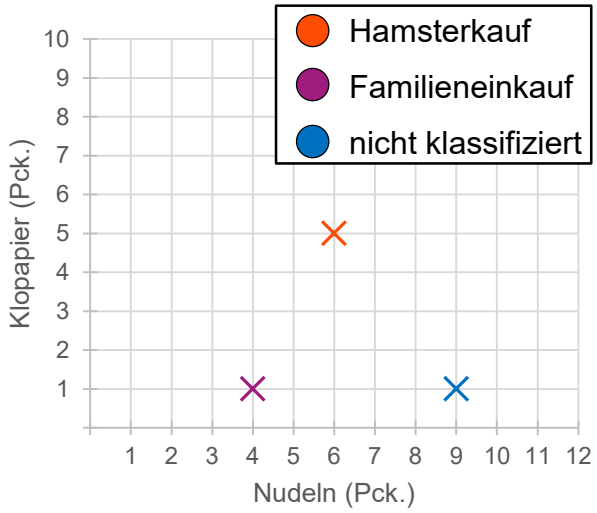
$$d(h,b) = \sqrt{(6-9)^2 + (5-1)^2} = \sqrt{9+16} = 5$$
$$d(f,b) = \sqrt{(4-9)^2 + (1-1)^2} = \sqrt{25+0} = 5$$

- Betragsmetrik:

$$d(h,b) = |6-9| + |5-1| = 3+4 = 7$$
$$d(f,b) = |4-9| + |1-1| = 5+0 = 5$$

- Maximumsmetrik:

$$d(h,b) = \max\{|6-9|, |5-1|\} = 4$$
$$d(f,b) = \max\{|4-9|, |1-1|\} = 5$$



Konstruiertes Beispiel: Einkaufswagen-Klassifikation II

Jeder Weg führt zu einer anderen Klassifikation des Einkaufswagens:

- Euklidische Metrik:

  $d(h, b) = 5$

  $d(f, b) = 5 \quad \Rightarrow$ Keine Entscheidung möglich.

- Betragsmetrik:

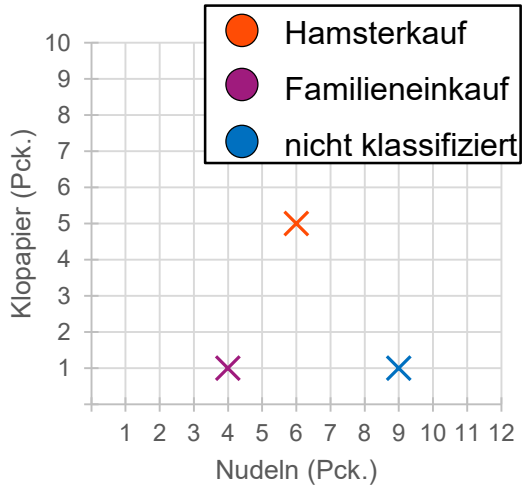
  $d(h, b) = 7$

  $d(f, b) = 5 \quad \Rightarrow$ Klassifik. als Familieneinkauf.

- Maximumsmetrik:

  $d(h, b) = 4$

  $d(f, b) = 5 \quad \Rightarrow$ Klassifik. als Hamsterkauf.



k Nearest Neighbor-Klassifikation: Formalisierung

Gegeben sei ein Datensatz mit n Beispielen mit zwei metrisch skalierten Features und einem Label in Form von Klassen.

Da die Features metrisch skaliert sind, kann jedes Beispiel mit einem im kartesischen Koordinatensystem eingefärbten Punkt repräsentiert werden. Diese Punkte werden mit $x_1, \ldots, x_n$ bezeichnet, wobei jeder Punkt in der Farbe seiner jeweiligen Klasse eingefärbt ist (bereits bekannter Scatterplot).

k Nearest Neighbor-Klassifikation: Formalisierung

Wird nun ein neues Beispiel betrachtet, sind die Ausprägungen der Features bekannt, das Beispiel kann als neuer Punkt b im kartesischen Koordinatensystem dargestellt werden. Die Klassifikation des Punktes erfolgt dann wie folgt:

- Auswahl der Metrik $d : M \times M \longrightarrow \mathbb{R}$.
- Bestimmung der k nächsten Nachbarn des Beispiels mittels Betrachtung von $d(x_i, b)$ für $i = 1, \dots, n$.
- Bestimmung der am häufigsten vertretenen Klasse unter den k Nachbarn.
- Klassifizierung des neuen Beispiels in diese Klasse.

k Nearest Neighbor-Klassifikation: Abstrakte Metriken

- Bei den bisher thematisierten Metriken handelt es sich um Metriken, welche zur Klassifikation von metrisch skalierten Features gut geeignet sind.

- Metriken können deutlich abstrakter sein, beispielsweise zur Klassifikation von Wörtern (bspw. Sprachassistenz, Spamerkennung, …)
Zwei Wörter sind Nachbarn, wenn möglichst viele Buchstaben gleich sind / möglichst viele Buchstaben an derselben Stelle stehen…

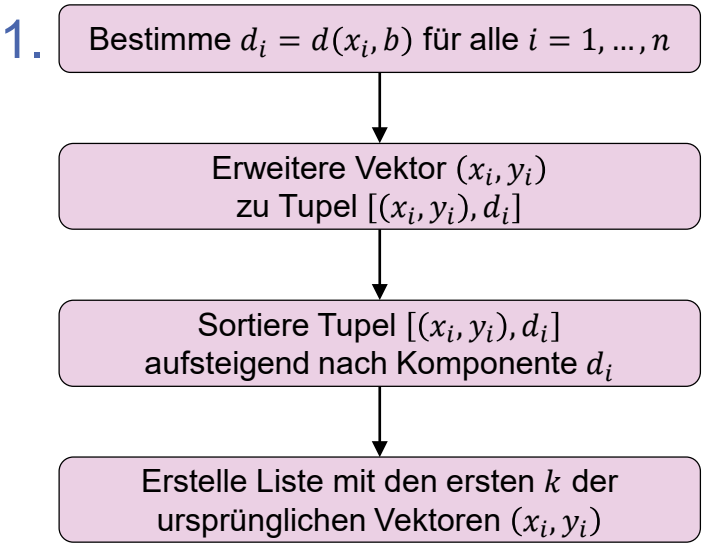
Datenverarbeitung I
Einkaufswagen-Klassifikation

Nächste Schritte:

- Starten der Anaconda Distribution.
- Öffnen des Jupyter Notebooks „Datenverarbeitung.kNN.ohneTB“.

k Nearest Neighbor-Klassifikation: Pseudocode

Ausgangssituation: Im Speicher liegen Vektoren (x_i, y_i), wobei x_i die Ausprägung der Features und y_i eine Kennzahl für die Klasse enthält, als Liste ab. Anzahl k der Nachbarn und Metrik d sind festgelegt.

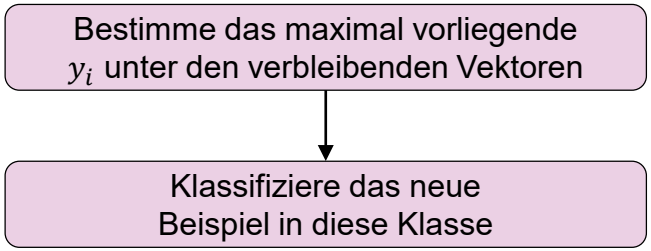



```python
# Anlegen der Funktion kNN_suche
def kNN_suche(X, neu, k):
    abstaende = list()
    for zeile in X:
        abstand = abstand_euklidisch(neu,zeile)
        abstaende.append((zeile, abstand))
    abstaende.sort(key=lambda tup: tup[1])
    print(abstaende)
    nachbarn = list()
    for i in range(k):
        nachbarn.append(abstaende[i][0])
    print(nachbarn)
    return nachbarn
```

k Nearest Neighbor-Klassifikation: Pseudocode

Ausgangssituation: Im Speicher liegen Vektoren (x_i, y_i), wobei x_i die Ausprägung der Features und y_i eine Kennzahl für die Klasse enthält, als Liste ab. Anzahl k der Nachbarn und Metrik d sind festgelegt.

2.

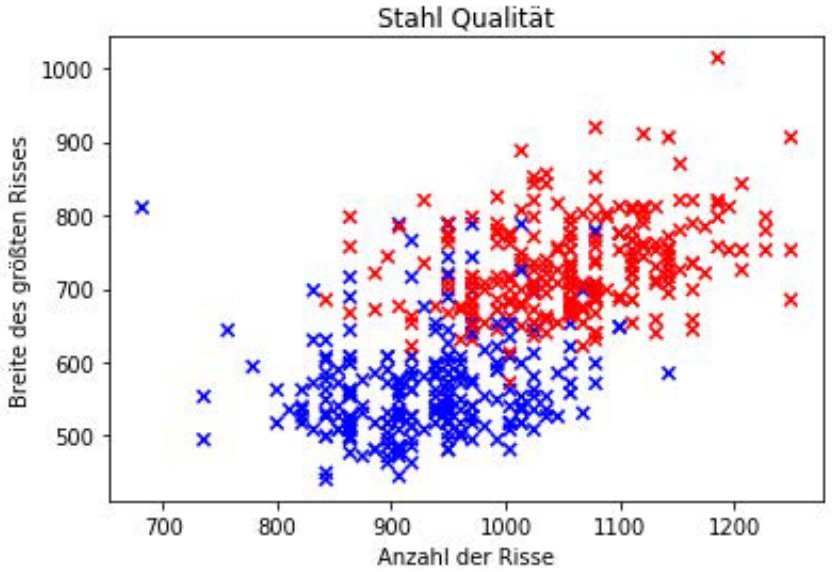



```python
# Anlegen der Funktion kNN_vorhersage
def kNN_vorhersage(X, neu, k):
    nachbarn = kNN_suche(X, neu, k)
    output = [row[-1] for row in nachbarn]
    vorhersage = max(set(output), key=output.count)
    return vorhersage
```

3.3 k Nearest Neighbor-Klassifikation mit scikit-learn

Stahlprojekt: Modellerstellung

Es wird nun erneut das Ausgangsbeispiel zur Stahlqualität, sowie die Möglichkeit der Bibliothek scikit-learn für Lernen in Python betrachtet.



Wiederholung: Modellerstellung überwachtes Lernen

- Mit dem überwachten Lernverfahren wird eine Modell (Zuordnung von Features zum Label) entwickelt.

- Trennung in Trainings- und Testdaten:
  - Mit Trainingsdaten wird das Modell entwickelt („gefittet“).
  - Mit Testdaten wird die Güte des Modells überprüft (Weicht das vom Modell vorhergesagte Label vom bereits bekannten Label in den Testdaten ab?).

Datenverarbeitung II
Stahlprojekt

Nächste Schritte:

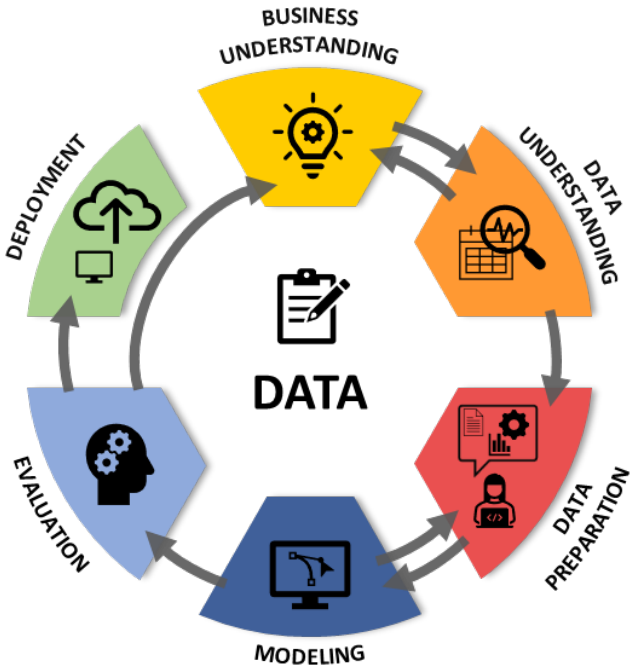
- Starten der Anaconda Distribution.
- Öffnen des Jupyter Notebooks „Datenverarbeitung.kNN.scikitlearn“.

Aufgabe zum Abschluss

Bearbeiten Sie mit Rückblick auf die Ausgangssituation und das heutige Vorgehen die folgenden zwei Arbeitsaufträge:

1. Fassen Sie in 2 Sätzen zusammen, was Sie heute am Ende der Veranstaltung in dem Projekt bereits erreicht haben.

2. Notieren Sie Pläne für weitere Datenerhebungen, Abläufe und Fragen, die Sie innerhalb des Projekts erwarten (nutzen Sie das CRISP-DM als Inspiration und ordnen Sie Ihre Pläne ggf. einem der Arbeitsbereiche zu).



Vielen Dank für Ihre Aufmerksamkeit!

Foto von Lenin Estrada von Pexels