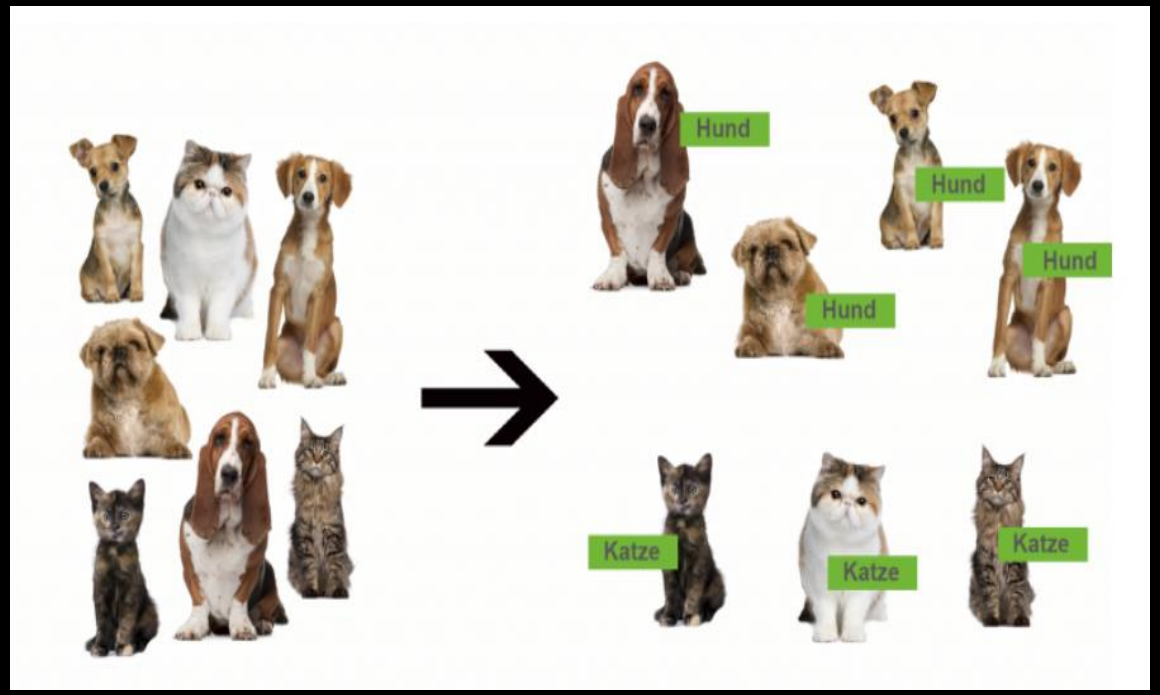
MACHINE LEARNING I

Vorlesung: Künstliche Intelligenz

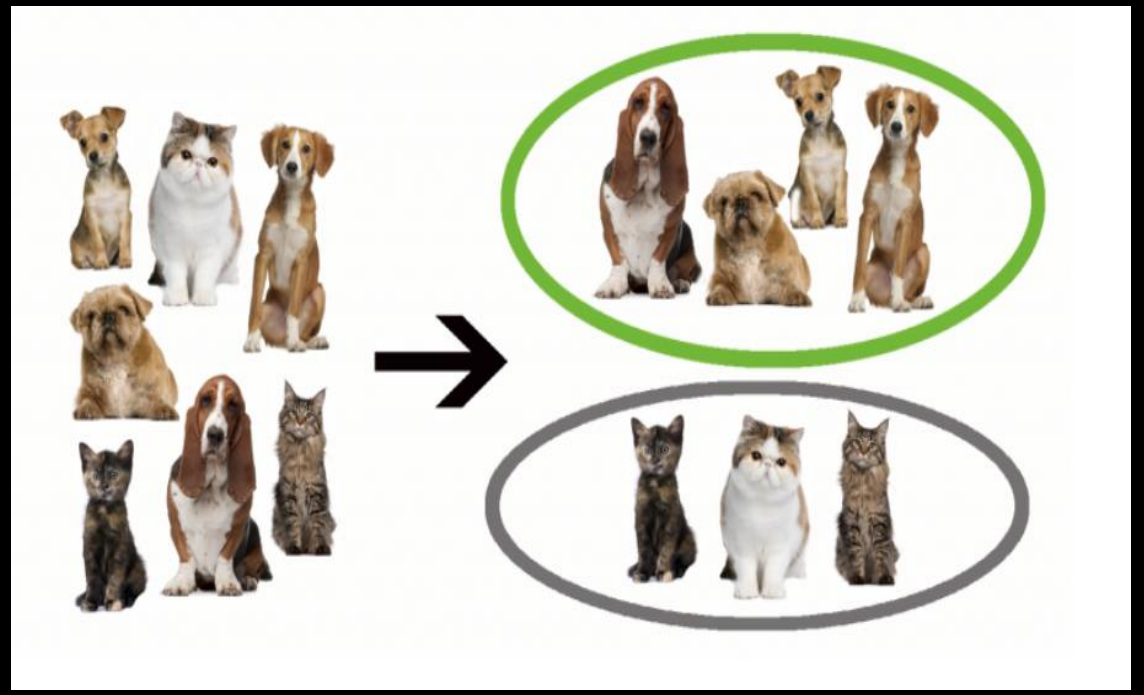
Kunal Oberoi, Matthias Tschöpe

LEARNING IN ML

Supervised Learning

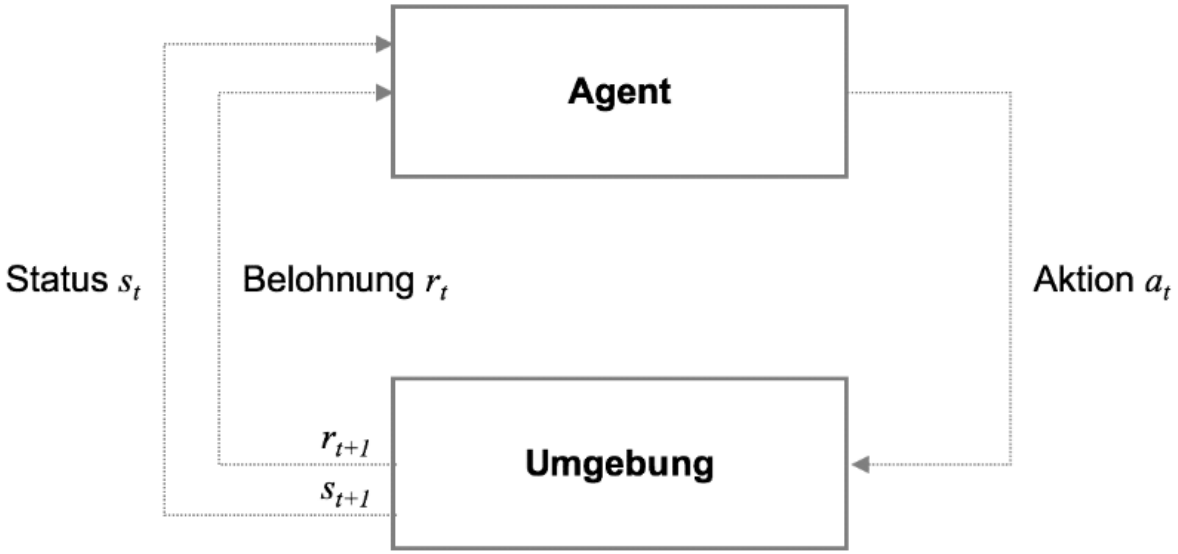


Unsupervised Learning



LEARNING IN ML

Reinforcement Learning



DATEN SPLITTEN

Trainingsdaten

- Trainingsdaten werden verwendet um unser Modell zu träinieren

Validationsdaten

- Validationsdaten dienen der Generalisierung, d.h. wie gut kann unser Modell lernen mit Daten die bisher noch nicht gesehen worden sind

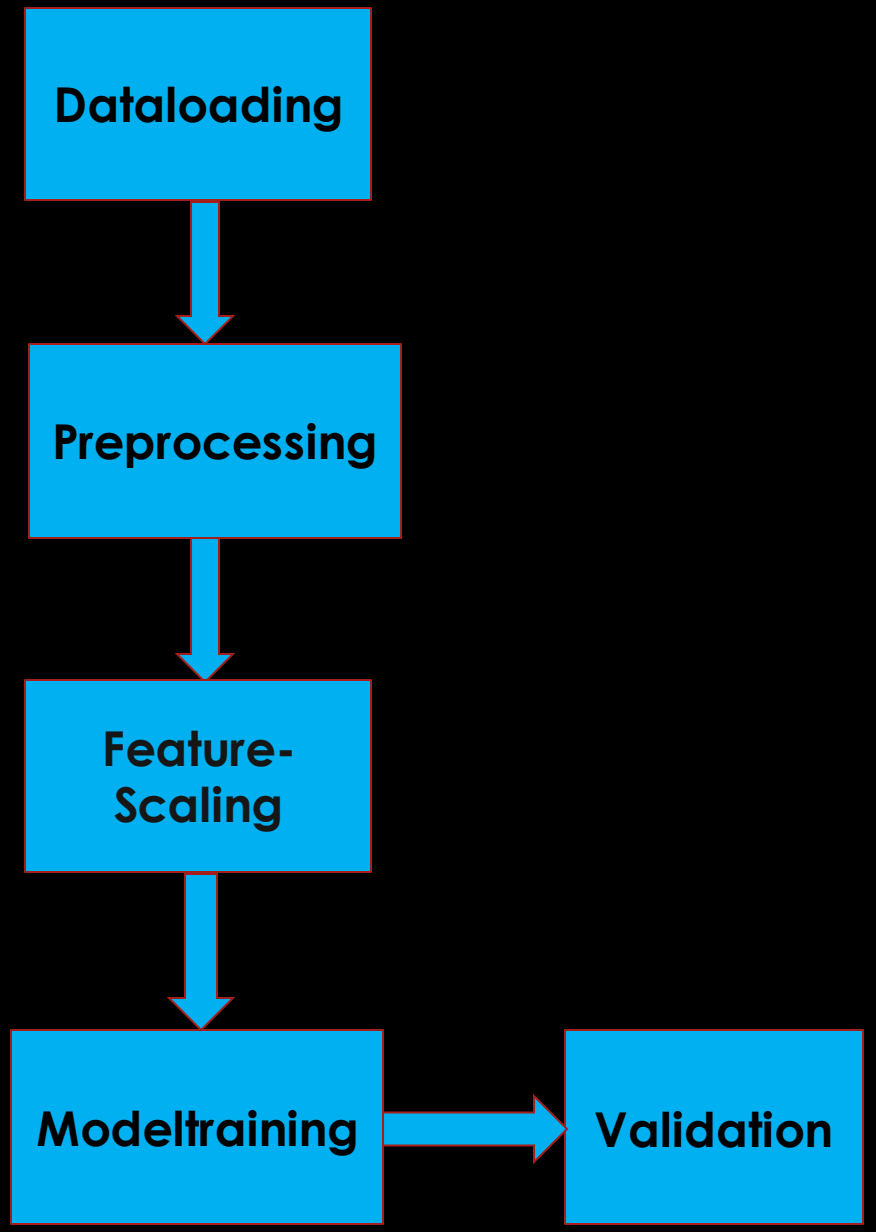
Testdaten

- Wenn das Modell vollständig optimiert und einsatzbereit ist, wird es anhand der Testdaten ausgewertet und wir bekommen unser Ergebnis wie gut das Modell ist.

ML - PIPELINE

Wie implementieren wir einen Klassifizierer?

1. **Erstens gibt es auch Pipelines in anderen bereichen z.B Computergrafik (Shader) und andererseits sieht die Pipeline hauptsächlich für Klassifizierer und ggf für Unsupervised Learning so aus.**
2. **Die Pipelines können unterschiedlich sein, wenn wir ein Reinforcement Problem betrachten**



SCIKIT-LEARN PACKAGE

- Sinnvoll für Daten Analyse
- Lineare Regression, Klassifikation und Clustering möglich etc.
- Verwendung:
    - pip install sklearn
    - Weiteres unter https://scikit-learn.org/stable/index.html#

BREAST CANCER DETECTION (ML-PROBLEM)

Die Ärtze identifizieren nicht jede Brustkrebspatientin, somit ist es ein Grund, warum Machine-Learning Ingenieure oder Data Scientist ins Bild kommen, weil sie das Problem durch die Mathematik und Computational Power lösen.

Zur Vereinfachung gehen wir davon aus, dass Sachbearbeiter der Ärtze uns bereits Feature aus MRT-Bilder extrahiert haben.

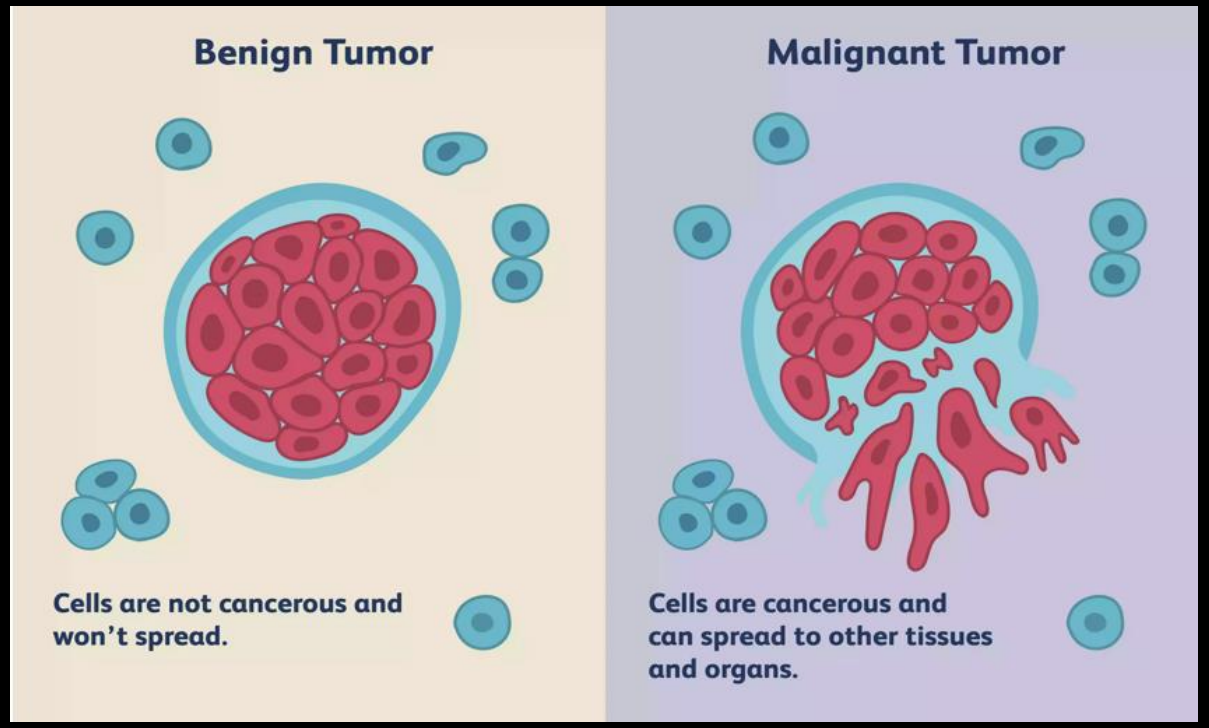
ÜBERBLICK DER DATEN

Die CSV-Datei besteht aus zweispalten ('malignant', 'benign')

Malignant = gefährlicher Tumor (1)

Benign = gutartiger Tumor (0)

wir laden die Daten direkt über Scikit-Learn



AUFGABE 1

- a) Laden Sie die **Daten** von Scikit-Learn über Folgenden den Befehl in ihr Python Notebook und speichern sie dies in eine Variable ab
  - from skearn.datasets import load_breast_cancer

- b) Erstellen Sie ein DataFrame um daten und target zu konkatinieren, zunächstmal brauchen Sie ein weiteres Package pip install pandas und numpy(bereits bekannt). Hinweis: pandas bietet eine Funktion DataFrame()
- c) **Visualisieren** sie ihre Daten über ein weiteres Package seaborn und verwenden sie die Methode pairplot()
- d) **Datapreprocessing**
  - Bereiten sie Ihre Input variable sowie die Outputvariable vor und speichern sie die Input variable X ab und Outputvariable als y

AUFGABE FORTSETZUNG

- e) In der Folie 4 haben Sie gesehen wie und warum man Datensplittet nun verwenden Sie die methode train_test_split() für die Daten in Trainings und Testdaten zu unterteilen. Hinweis: from sklearn.model_selection import train_test_split, in der Form x_train,x_test,y_train,y_test = train_test_split()
- f) **Feature Scaling oder Normalisieren**
  - Konvertieren sie die verschiednen Einheiten und Größenangaben in einer Einheit
    - Hinweis: from sklearn.preprocessing import StandardScaler und verwenden sie die Methode fit_transform() und transform()

    - **Nun sind sie soweit das die Daten bereit sind und wir sie durch ML-Modelle Träinieren können**
    - Sie können schon folgende packages importieren from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

MODELLE – ML (KURZER-ÜBERBLICK)

Support Vector Klassifier

Das Ziel des Support Vector Machine-Algorithmus besteht darin, eine Hyperebene in einem N-dimensionalen Raum (N - die Anzahl der Merkmale) zu finden, die die Datenpunkte eindeutig klassifiziert

Daten werden in Zweiklassen separiert, in unserem Fall in Malignant und Benign

K- Nearest Neighbor Classifier

Der K-nearest-Neighbour-Algorithmus ist ein Klassifikationsverfahren, bei dem eine Klassenzuordnung unter Berücksichtigung seiner k-nächsten Nachbarn vorgenommen wird.

Decision Tree Classifier

Ein Entscheidungsbaum Klassifizierer ist Art Flussdiagram zu verstehen. Die inneren Knoten sind Zwischenstufen die jeweils einem Bereich des Features zugeordnet sind. Jedes Blatt repräsentiert eine Region, die einer bestimmten Klasse zugeordnet ist

AUFGABE 2 (MODELLE-TRÄINIEREN)

- Verwenden Sie folgende Struktur für die in der Folie 11 genannten Modelle. Zunächst mal mit Daten ohne Skalierung und danach mit skalierten Daten, welche unterschiede erkennen sie und welches Modell ist das beste?

```
from sklearn.svm import SVC
svc_classifier = SVC()
svc_classifier.fit(...)
predicted_svm= svc_classifier.predict(...)
accuracy_score(...)
```

- Machen Sie das für alle 3 Modelle!

WAS IST EINE CONFUSION-MATRIX ?

Überblick

- Die Confusion-Matrix stellt eine Zusammenfassung der vorhergesagten Ergebnisse für ein Klassifizierungsproblem
- Anzahl der richtigen und falschen Vorhersagen werden mit Zählwerten zusammengefasst und nach Klassen aufgeteilt.
- Des Weiteren zeigt die Confusion-Matrix die Art und Weise, in der ihr Modell bei vorhersagen verwirrt ist.
- Verschafft nicht nur Einblick in die Fehler, die von einem Klassifizierer gemacht werden, sondern unteranderem auch die Art der Fehler

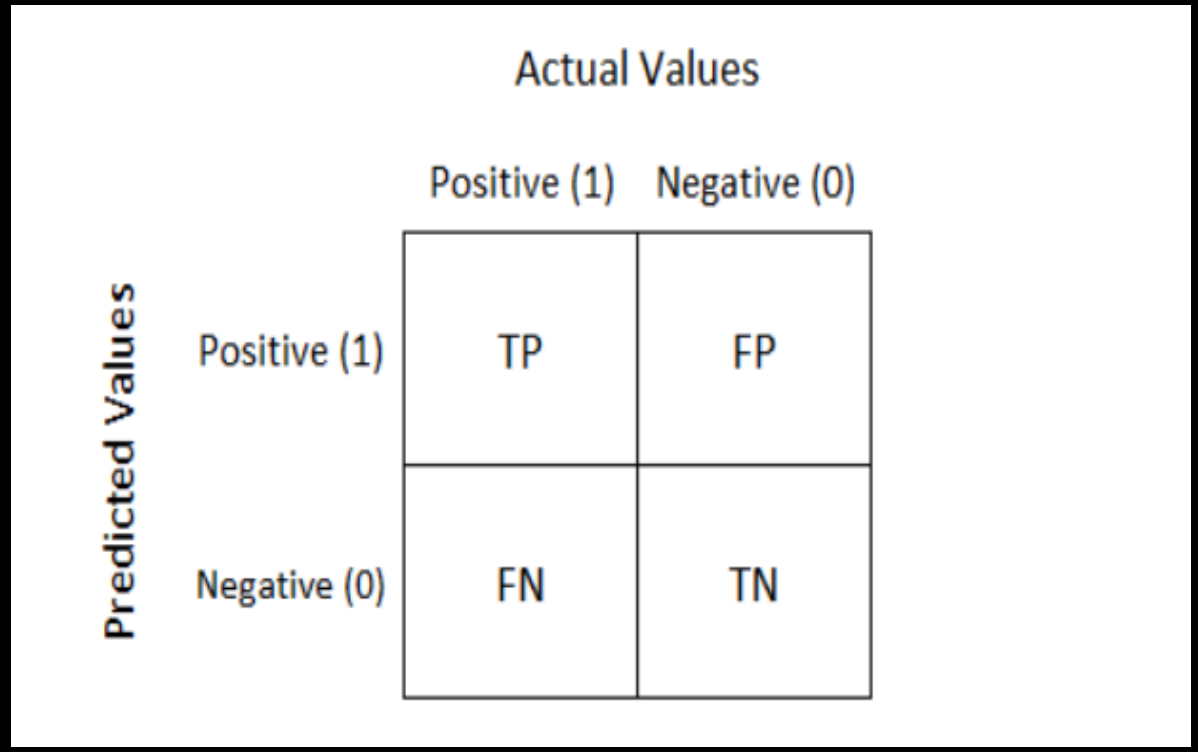
Begrifflichkeiten

- (P) Positiv: Vorhersage ist positiv z. B (wurde das Auto richtig erkannt)
- (N) Negativ: Vorhersage ist negativ z.B (kein Auto)
- (TP) True Positive: Alle Punkte die als eine Klasse erfasst wurden und wirklich zu dieser Klasse gehören.
- (FP) False Positive: Punkte, die als eine Klasse erfasst wurden, aber nicht zu ihr gehören.
- (FN) False Negative: Punkte, die nicht als eine Klasse erfasst wurden, aber zu ihr gehören.
- (TN) True Negative: Punkte, die nicht als eine Klasse erfasst wurden, und auch nicht zu ihr gehören

Recall und Precision

- Recall: Verhältnis der Gesamtzahl der richtig klassifzierten positiven Beispiele geteilt durch die Gesamtzahl der positiven Beipspiele. (Hoher wert von Recall zeigt, dass die Klasse richtig erfasst wurde, d.h geringe Anzahl von FN)
- Precision: Teilen die Gesamtzahl der richtig klassifizierten positiven Beispiele durch die Gesamtzahl der vorhergesagten positiven Beispiele. (Hoher wert von Precision zeigt, dass ein als positives gekennzeichnetes Beispiel tatsächlich positiv ist, d.h geringe Anzahl der FP)

AUFBAU DER CONFUSION MATRIX UND FORMEL PRECISION UND RECALL



$$\textit{Recall} = \frac{TP}{TP + FN}$$

$$\textit{Precision} = \frac{TP}{TP + FP}$$

AUFGABE 3 (CONFUSION-MATRIX UND RESULT)

- a) Erstellen Sie eine Confusion-Matrix mit Hilfe von Python für das Modell, dass die beste Genauigkeit liefert um die Daten zu Klassifzieren. Die Methode dazu exisitiert bereits confusion_matrix().
  - Hinweis: importieren sie matplotlib um die Matrix zu plotten,falls es vergessen wurde
  - Diskutieren Sie die Ergebnisse der **Confusion-Matrix**!
- b) Um die Ergebnisse zum Abschluss zu visualiseren erstellen Sie mit hilfe von classification_report() ein Endergebnis. Interpretieren Sie diese ! Alternativ auch für andere Modelle ausprobieren!

AUFGABE 4 (CROSS-VALIDATION)

Was ist Crossvalidation?

- Eine Technik der Leistung eines Vorhersahemodells

Wofür braucht man das?

- Um den Generalisierungsfehler zu bestimmen brauchen wir Daten, die beim Lernen nicht gesehen wurden. Bei Bedarf müssen mehr Daten aufgenommen werden

Aufgabe

Importieren Sie zunächst mit Hilfe von Scikit-Learn cross_val_score, Hinweis:from sklearn.model_selection import cross_val_score , füllen Sie das Model mit den cross_val_score(estimator=…, X=…,y=.., cv=2) und printen Sie Ihre Ergebnis. Wichtig ist speichern sie cross_val_score in eine Variable ab.