Machine Learning Tag 1 – Tagesaufgaben

1. Finde und beschreibe für jede der drei Machine-Learning-Klassen (supervised,

unsupervised, reinforcement) ein Beispiel. Warum ist ML hier der klassischen

Programmierung überlegen?

* Supervised Learning (Überwachtes Lernen)
* Ein klassischer Chatbot fällt meist ins überwachte Lernen, weil er auf gelabelten Dialogpaaren (Frage → Antwort) trainiert wird.
* Moderne generative Chatbots (z. B. GPT-Modelle) nutzen dagegen:
* Unüberwachtes Vortraining auf riesigen Textkorpora (Sprachmodellierung),
* Überwachtes Feintuning auf spezifischen Frage-Antwort-Datensätzen,
* Bestärkendes Lernen, um die Antwortqualität anhand menschlichen Feedbacks weiter zu optimieren.
* Kurz gesagt:
* **klassischer-Chatbots:** überwacht
* **Generative LLM-Chatbots:** eine Kombination aus unüberwacht (Pretraining), überwacht (Feintuning) und Reinforcement Learning
* Unsupervised Learning (Unüberwachtes Lernen)
* **Markt-Basket-Analyse (Association Rules)**  
  Entdeckung von Produktkombinationen, die häufig zusammen gekauft werden, zb. für Cross-Selling in Supermärkten.
* Reinforcement Learning (Bestärkendes Lernen)
* **Roboternavigation**  
  Navigation für autonome Fahrzeuge oder Drohnen in unbekannten oder sich verändernden Umgebungen.
* In allen Fällen ermöglichen ML-Methoden Lösungen, die klassische, regelbasierte Ansätze ohne enorme manuelle Regeldefinition und Anpassung erst gar nicht bewältigen könnten.

2. Finde zu den drei Machine-Learning-Klassen jeweils ein Youtube-Video und

schreibe Dir die wichtigsten Aussagen dazu auf.

* Video zum „Supervised Learning“:
* <https://www.youtube.com/watch?v=BkDUDi6YDaU&t=327s>
* Definition und Ziel  
  Supervised Learning ist eine Machine-Learning-Methode, bei der Modelle aus gelabelten Beispielen (Eingabe → Zielvariable) Muster lernen, um auf neue, unbekannte Daten Vorhersagen zu treffen.
* Beispielanwendung: Direktmarketing-Kampagne  
  Anhand von Bankkundendaten (Alter, Job, Kontostand etc.) wird ein Klassifikator trainiert, der vorhersagt, ob jemand auf eine Marketing-Aktion („yes“/“no“) reagiert.
* Trainings- und Testdaten  
  Das Dataset wird in Training und Test aufgeteilt. Nur das Trainingsset erhält der Algorithmus zum Lernen; das Testset dient dazu, die Generalisierungsfähigkeit (also die Performance auf unbekannten Daten) objektiv zu prüfen.
* Modelle und Algorithmen  
  Häufig genutzte Verfahren sind etwa logistische Regression, Random Forest oder XGBoost. Sie berechnen Wahrscheinlichkeiten für jede Klasse und entscheiden so über die Zuordnung.
* Overfitting vs. Generalisierung  
  Overfitting entsteht, wenn ein Modell das Trainingsset zu genau abbildet und auf neuen Daten schlecht performt. Durch Methoden wie Test-Split und geeignete Regularisierung wird ein gutes Gleichgewicht („guter Fit“) angestrebt.
* Video zum „Unsupervised Learning“:
* <https://www.youtube.com/watch?v=XSOi9MKfEHQ>
* Definition ohne Zielvariable  
  Unüberwachtes Lernen unterscheidet sich vom Supervised Learning dadurch, dass kein Label bzw. keine Zielvariable vorgegeben wird. Der Algorithmus durchsucht die Rohdaten selbstständig nach Mustern, Strukturen und Zusammenhängen.
* Hauptverfahren  
  Clustering (Clusteranalyse): Gruppierung in homogene Segmente  
  Assoziationsanalyse: Entdeckung von Regelmäßigkeiten (z. B. Warenkorbanalyse)  
  Dimensionsreduktion: Reduktion auf wenige Dimensionen (z. B. PCA, t-SNE) zur Visualisierung großer Datensätze
* Beispiel Bank-Direktmarketing  
  Am selben Bankdatensatz wie beim Supervised-Beispiel wird nun die Zielvariable ignoriert. Stattdessen versucht ein Cluster-Algorithmus (z. B. k-Means) eigenständig, auf Basis von Merkmalen wie Alter und Kontostand sinnvolle Kundengruppen zu bilden.
* Prozessablauf

Datenvorbereitung: Auswahl relevanter Prädiktoren

Cluster-Anzahl festlegen: Manchmal per Elbow-Methode o. ä.

Modellanwendung: Algorithmus segmentiert die Daten

Interpretation: Analysieren, welche Charakteristika die einzelnen Cluster auszeichnen

* Nutzen und Anwendung  
  Die gebildeten Segmente können für zielgerichtete Marketingkampagnen, Produktentwicklung oder Prozessoptimierungen genutzt werden, ohne dass zuvor manuell Regeln definiert werden müssen.
* Video zum „Reinforcement Learning“:
* <https://www.youtube.com/watch?v=EAX12jlMlUw>
* Grundprinzip: Trial-and-Error in einer Umgebung  
  Ein Agent interagiert mit einer abgegrenzten Umgebung. Durch Ausprobieren von Aktionen erhält er sukzessive Belohnungen (Rewards) und lernt so, welche Aktionen in welchen Zuständen (States) am erfolgversprechendsten sind.
* Formale Komponenten eines RL-Problems
* Zustände (S): Beschreiben die aktuelle Situation (z. B. Position des Saugroboters und Verteilung des Schmutzes).
* Aktionen (A): Mögliche Schritte des Agenten (z. B. bewegen oder reinigen).
* Policy (π): Strategie des Agenten, die angibt, mit welcher Wahrscheinlichkeit er in Zustand s Aktion a wählt (π(a∣s)\pi(a|s)π(a∣s)).
* Belohnungsfunktion (R): Ordnet jedem Übergang (s, a → s′) einen Reward zu, um erwünschtes Verhalten zu fördern.
* Beispiel „Saugroboter“  
  In einem 5×5-Grid soll der Roboter schmutzige Felder säubern und zur Ladestation zurückkehren. Jede Aktion (Bewegung oder Reinigen) ändert den Zustand und liefert einen Reward (z. B. +100 fürs Reinigen, –1 fürs Bewegen, –100 für Kollision mit Wand), sodass über hunderte Durchläufe eine effiziente Reinigungs- und Rückkehr-Strategie entsteht.
* Modellbasiert vs. Modellfrei
* Modellbasierte Verfahren bauen explizit auf einer MDP-Darstellung (Markov Decision Process) auf und nutzen Algorithmen wie Policy-Iteration oder Value-Iteration.
* Modellfreie Verfahren (z. B. Q-Learning, Policy-Gradient-Verfahren) benötigen kein vollständiges Umweltmodell und optimieren direkt Policy oder Wertfunktion.
* Fortgeschrittene Ansätze: Deep RL und Policy-Optimierung  
  Mit neuronalen Netzen (Deep RL) lassen sich auch hochdimensionale Zustände verarbeiten. Methoden wie PPO, DQN oder Actor-Critic kombinieren Policy-Gradient-Ansätze und Wert-Schätzung, um in komplexen, nicht-linearen Umgebungen robuste Strategien zu erlernen.

3. Recherchiere und beschreibe in deinen eigenen Worten diese 5 Problemstellungen

des maschinellen Lernens:

Regression

Man kann sich Regression vor wie das Zeichnen einer Geraden durch viele Punkte in einem Diagramm vorstellen, so dass die Gerade möglichst nah an allen Punkten liegt.

* Du hast Eingabewerte (z. B. Alter, Zimmerzahl, Lage eines Hauses)

und willst damit eine Zahl vorhersagen (z. B. den Preis des Hauses).

* Der Algorithmus sucht die beste Linie (oder Kurve), die erklärt, wie sich der Preis mit den Eingaben ändert.
* Er passt diese Linie so an, dass die Abstände der Punkte zur Linie insgesamt möglichst klein sind (also die Vorhersagen möglichst genau).
* Auf diese Weise kann er für neue Häuser anhand ihrer Merkmale schnell einen realistischen Preis abschätzen.

Klassifizierung

Stell dir vor, du möchtest deine E-Mails automatisch in „Spam“ oder „Nicht-Spam“ sortieren / labeln.

* Was passiert?  
  Ein Programm schaut sich jede E-Mail an und entscheidet, ob sie in „Spam“ oder „Nicht-Spam“ gehört.
* So läuft’s ab:

Du gibst dem Programm viele Beispiele, die bereits richtig einsortiert sind (gelabelte Daten).

Es lernt daraus, welche Merkmale typisch für jede Klasse sind (z. B. bestimmte Wörter im Betreff).

Bei neuen E-Mails wendet es dieses Gelernte an und wählt die passende Klasse.

* Kurz: Klassifizierung bedeutet, Objekte in eine von mehreren vorgegebenen Gruppen einzuteilen – basierend auf dem, was das Modell in der Vergangenheit gelernt hat.

Clustering

Stell dir vor, du sortierst eine Kiste voller bunter Kugeln ganz von allein nach Farbe und Größe – ohne dass dir jemand sagt, welche Kugel wohin gehört.

* Was passiert?  
  Ein Programm schaut sich alle Kugeln (oder Datensätze) an und legt Gruppen (Cluster) an, in denen die Objekte einander möglichst ähnlich sind.
* Wie läuft das ab?

Es misst „Ähnlichkeit“ (z. B. Farbe, Größe, Gewicht).

Es bildet automatisch Gruppen, in denen die Unterschiede zwischen den Gruppen möglichst groß und innerhalb einer Gruppe möglichst klein sind.

Natural Language Processing

Stell dir vor, du bringst einem Computer bei, Texte und gesprochene Sprache so zu verstehen, wie wir Menschen es tun. NLP ist also die Technik, mit der Programme:

* Wörter in kleinere Einheiten (Tokens) zerlegen,
* erkennen, welche Wortart (Nomen, Verb, …) ein Wort ist,
* herausfinden, ob ein Text positiv oder negativ klingt (Sentiment),
* Texte in eine andere Sprache übersetzen oder
* selbst neue Sätze schreiben können (z. B. Chatbots).
* Kurz: NLP hilft Computern, menschliche Sprache zu lesen, zu verstehen und selbst zu erzeugen.

Hauptkomponentenanalyse (PCA)

Stell dir vor, du hast sehr viele Messwerte (zum Beispiel 20 verschiedene Sensoren), möchtest aber nur die wichtigsten Informationen behalten. PCA fasst diese vielen Werte zu wenigen „Superwerten“ zusammen, die möglichst viel von der ursprünglichen Variantenvielfalt erklären.

* Du rechnest aus, welche Kombinationen der Originalwerte die größte Streuung (Varianz) haben.
* Dann behältst du nur die ersten Komponenten dieser Kombinationen („Hauptkomponenten“).
* So wird dein Datensatz übersichtlicher, Rauschen verschwindet und spätere Analysen (z. B. mit ML-Algorithmen) laufen schneller und oft besser.

4. Beschreibe in eigenen Worten, warum Overfitting ein Problem im Machine

Learning ist und wie dem entgegengewirkt werden kann.

Overfitting entsteht, wenn ein Modell die Trainingsdaten zu genau abbildet – es lernt auch Rauschen und zufällige Details – und dadurch auf neuen Daten schlecht generalisiert. Ein überfittetes Modell zeigt exzellente Performance auf den Trainingsdaten, versagt jedoch bei unbekannten Daten, weil es keine verallgemeinerbaren Muster erfasst.

Gegenmaßnahmen:

* Train-Test-Split & Kreuz­validierung: Bewertung auf einem separaten Datensatz oder mittels Kreuzvalidierung verhindert zu optimistische Fehlereinschätzungen.
* Regularisierung: Bestraft zu große Modellkomplexität und hält Parameterwerte klein.
* Early Stopping: Stoppt das Training, sobald die Validierungsfehler nicht mehr sinken.
* Ensemble-Methoden: Kombination mehrerer schwächerer Modelle reduziert Varianz und Überanpassung.

5. Was sind Hyperparameter? Recherchiere, wie man die besten Hyperparameter

findet!

Hyperparameter sind Einstellungen, die den Lernprozess eines Modells steuern (z. B. Lernrate, Anzahl der Bäume in einem Random Forest oder Regularisierungsstärke). Sie werden nicht im Training selbst gelernt, sondern vorab festgelegt.

Häufige Tuning-Verfahren:

* Grid Search: Exhaustives Durchsuchen eines vordefinierten Rasters von Parameterkombinationen, bewertet über Kreuzvalidierung.
* Random Search: Zufällige Auswahl von Kombinationen; oft effizienter, wenn wenige Hyperparameter wirklich entscheidend sind.
* Bayesian Optimization: Baut ein probabilistisches Modell der Ziel–Funktion (z. B. Validierungsfehler) auf und wählt iterativ vielversprechende Einstellungen aus.
* Gradientenbasierte Methoden & Evolutionäre Algorithmen: Optimieren Hyperparameter direkt mittels Gradienten oder evolutionärer Strategien, oft in komplexen, hochdimensionalen Räumen.

Durch sorgfältiges Hyperparameter-Tuning lässt sich die Modellleistung signifikant steigern und eine bessere Generalisierung erreichen.