**Algorithmus für Empfehlungen**

**User Profiling**

* Nutzerdaten werden gesammelt
* Unter anderem Demografik, Interaktionen, Historie, Verhaltensmuster

**Content Analysis**

* Inhalte werden analysiert
* Metadaten werden den Inhalten zugeteilt
* Inhalte werden nach Themengebiete, Relevanz, usw. kategorisiert

**Collaborative Filtering**

* Nutzerverhalten werden analysiert und mit anderen Nutzern verglichen
* Inhalte werden empfohlen, die ähnlichen Nutzern gefällt oder bei der interagiert wurde

**Content-Based Filtering**

* Empfehlungen anhand von zuvor gesehenen Inhalten, die dem Nutzer gefallen hat
* Metadaten der Inhalte werden verglichen

**Machine Learning-based Models**

* KI wird trainiert und analysiert Nutzerinteraktionen, Historie und Präferenzen
* Passende Empfehlungen werden von der KI vorhergesagt

**Natural Language Processing (NLP) Techniques**

* KI wird mit Texten aus Inhalten trainiert
* Kontext und Empfindungen werden von der KI analysiert, um die Präferenzen zu verstehen
* KI empfiehlt Inhalte anhand der Präferenzen

**Reinforcement Learning/Feedback Loop**

* Nutzerfeedback werden analysiert
* Empfehlungen werden anhand des Feedbacks optimiert

**Hybrid Approaches**

* Kombination aus unterschiedlichen Methoden, um Empfehlungen zu optimieren

**Real-Time Updates**

* Empfehlungen werden unmittelbar aktualisiert
* Änderungen in den Nutzerpräferenzen werden sofort berücksichtigt

**Diversity and Serendipity**

* Empfehlungen werden manchmal unabhängig der Präferenzen gezeigt
* stellt dem Nutzer neue, andere Inhalte und Themen vor
* verhindert eine “filter-bubble”

**Proof of Concept für Empfehlungen**

Personalisierte Empfehlungen sollen anhand von Nutzerdaten in der App angezeigt werden. Die Nutzerdaten beinhalten den Standort, präferierte Brettspielgenres und -arten, Alter der Person, Erfahrungsgrad, und zuvor aufgerufene Inhalte. Für die Kategorisierung der Inhalte werden diese mit Metadaten versehen. Die Metadaten der Brettspiele beinhalten Brettspielgenres und -arten, Altersempfehlung, Schwierigkeitsgrad, Bewertungen und Anzahl der Aufrufe. Für die Gruppensuche werden die Metadaten Standort, Brettspielgenres und -arten, Alter für die Veranstaltung und Erfahrungsgrad benötigt.

Für das PoC werden Beispieldaten der Brettspiele erstellt. Für den Test des Algorithmus sollen die Methoden Collaborative Filtering und Content-Based Filtering zum Einsatz kommen. Collaborative Filtering wird benutzt, um Inhalte von Nutzern mit ähnlichen Präferenzen zu empfehlen. Content-Based Filtering wird hingegen benutzt, um ähnliche Inhalte zu empfehlen.

Es soll für den Prototyp eine einfache Benutzeroberfläche im Android Studio und ein Backend erstellt werden. Die Datenbank wird mithilfe von MongoDB erstellt.

In der Datenbank sollen die Nutzerdaten und Brettspiele hinterlegt werden.

Nutzer sollen auf der Benutzeroberfläche Gruppensuche und Brettspiele aufrufen und ihre Präferenzen bearbeiten können.

Interaktionen der Nutzer werden in der Datenbank aktualisiert.

Exit-Kriterien:

* Inhalte, die ähnlichen Nutzern gefielen werden empfohlen und auf der Benutzeroberfläche angezeigt
* ähnliche Brettspiele werden empfohlen und auf der Benutzeroberfläche angezeigt, wenn ein Brettspiel aufgerufen wird

Fail-Kriterien:

* Inhalte, die nicht zu den Nutzer passen werden empfohlen
* ähnliche Inhalte, die nicht zu den Inhalt passen werden empfohlen
* keine Inhalte werden empfohlen

Fallbacks:

* der Nutzer soll nicht-relevante Inhalte als solches markieren können, um den Algorithmus zu verbessern