reínforcement learning, multui armed bandit, ' use case doporučování filmů nebo něco podobného v půjčovně filmů jako je Netflix či Oneplay

**ChatGPT řekl:**

Skvělá volba tématu! **Reinforcement Learning (RL)** a **Multi-Armed Bandit (MAB)** modely se velmi hodí pro **doporučovací systémy** ve službách jako **Netflix, OnePlay, HBO Max, apod.** Níže ti popíšu, jak se tyto koncepty aplikují v praxi, konkrétně v prostředí filmové půjčovny.

**🎯 Use Case: Doporučování filmů pomocí Multi-Armed Bandit v půjčovně filmů**

**🧠 Kontext**

Platforma jako Netflix má obrovskou knihovnu filmů a seriálů. Uživatelům musí doporučit obsah, který:

* je pravděpodobně zaujme,
* zvýší míru sledování a spokojenosti,
* maximalizuje retenci uživatelů.

**🔄 Co je Multi-Armed Bandit (MAB)?**

MAB je typ problému v RL, kde máš několik „pák“ (např. filmy) a pokaždé si vybereš jednu. Cílem je **maximalizovat dlouhodobou odměnu**, např. kolik uživatelů daný film sleduje po doporučení.

**Příklad paralely:**

* 🎰 Každý film = jedna „páka“ na výherním automatu.
* 👤 Uživatel = hráč, kterému doporučíme jednu „páku“ (film).
* 🎁 Odměna = jestli si film pustil, jak dlouho ho sledoval, jestli ho ohodnotil kladně, atd.

**🛠️ Jak MAB funguje v praxi?**

**1. Zásadní výzva: Exploration vs. Exploitation**

* **Exploitation**: Doporučím film, který v minulosti fungoval dobře.
* **Exploration**: Vyzkouším méně známý film, který by mohl fungovat lépe.

MAB algoritmy (např. ε-greedy, UCB, Thompson Sampling) tento problém řeší automaticky.

**🔎 Use Case - Příklad:**

**👤 Uživatelský scénář:**

1. Uživatel otevře aplikaci (např. OnePlay).
2. Algoritmus má připravených 10 kandidátů (filmů), které připadají v úvahu.
3. MAB algoritmus vybere jeden z nich.
4. Sleduje se, jestli:
   * si ho pustil (binární odměna 0/1),
   * jak dlouho ho sledoval,
   * dal hodnocení, apod.

Na základě toho se upravují „odměny“ pro jednotlivé filmy v čase.

**📊 Příklady konkrétních MAB algoritmů:**

| **Algoritmus** | **Popis** |
| --- | --- |
| **ε-greedy** | Většinou vybírá nejlepší známou možnost, ale někdy náhodně zkouší novou. |
| **UCB (Upper Confidence Bound)** | Bere v potaz nejen průměrnou odměnu, ale i nejistotu. |
| **Thompson Sampling** | Pravděpodobnostní přístup – vzorkuje z rozdělení odměn a vybírá tu nejlepší. |

**🧪 Výhody použití MAB v doporučování filmů:**

* **Online learning**: Průběžně se učí z nových dat.
* **Personalizace**: Může se přizpůsobit každému uživateli (contextual bandits).
* **Efektivita**: Rychle adaptuje strategii doporučování bez potřeby velkého batch tréninku.

**🎬 Možné rozšíření – Contextual Bandits:**

Pokud přidáš **kontext o uživateli** (např. věk, historie sledování, zařízení, denní doba), můžeš použít **Contextual Bandit** modely, které dělají personalizovaná rozhodnutí.

**💡 Shrnutí:**

| **Oblast** | **Detail** |
| --- | --- |
| **Použití** | Doporučení filmů v reálném čase |
| **Model** | Multi-Armed Bandit |
| **Metoda** | ε-greedy, UCB, Thompson Sampling |
| **Výhoda** | Průběžné učení, online adaptace, řešení exploration/exploitation |
| **Rozšíření** | Contextual Bandits (pro personalizaci) |

Pokud chceš, mohu ti také pomoci s jednoduchým **Python skriptem** na simulaci doporučování filmů pomocí MAB, nebo vytvořit **prezentační slidy** či vizualizaci fungování algoritmu.

Chceš něco z toho?