

# Legal Case Predictor

Cieľom tohto projektu bolo vytvoriť program, ktorý dokáže analyzovať nový právny prípad a porovnať ho s databázou už zaznamenaných prípadov. Na základe podobnosti textu program vyhledá najrelevantnejšie minulé prípady a následne využije OpenAI na návrh ďalších krokov a predikciu možného rozsudku.

Projekt spája základnú analýzu textu s umelou inteligenciou a predstavuje ukážku, ako by mohli fungovať systémy na podporu právnych rozhodnutí.

Vzhľadom na obmedzený prístup ku Codexis som použil simulovanú databázu právnych prípadov v textovom súbore **cases.txt**. Táto databáza obsahuje fiktívne prípady, ktoré umožnili demonštrovať funkcionality programu a overiť algoritmus vyhľadávania podobnosti a generovania promptu pre AI.

## Funkcionalita programu

### 1. Príprava prostredia

- Uistite sa, že máte nainštalovaný Python (verzia 3.10 alebo vyššia).
- Nainštalujte knižnicu openai pomocou príkazu:

```
pip install openai
```

### 2. Nastavenie API kľúča

- Otvorte súbor **main.py**.
- Nastavte premennú OPENAI\_API\_KEY na váš OpenAI API kľúč:

```
OPENAI_API_KEY = "YOUR_API_KEY_HERE"
```

- OpenAI API kľúč je možné získať na tejto stránke: <https://platform.openai.com>

### 3. Príklady a testovacie súbory

- V GitHub repozitári je priložený textový súbor cases.txt, ktorý obsahuje simulovanú databázu 30 právnych prípadov používaných programom na vyhľadávanie podobností.
- Pre jednoduchšie testovanie je možné priložiť aj ďalší textový súbor sample\_inputs.txt obsahujúci niekoľko ukážkových vstupov, ktoré môžu byť skopírované a použité priamo do programu, aby sa nemuselo každý krát manuálne zadávať dlhý popis prípadu.
- Program potom pre každý vstup vypíše odporúčané ďalšie kroky a predikciu rozhodnutia súdu na základe najpodobnejších prípadov z databázy.

#### 4. Spustenie programu

- Program sa spúšťa cez terminál alebo príkazový riadok:

```
python main.py "Popis vášho nového právneho prípadu"
```

- Popis je potrebné zadať v češtine, aby sa zachovala kompatibilita s textami v databáze.
- Program následne vypíše odporúčané ďalšie kroky a predikciu rozhodnutia súdu na základe najpodobnejších prípadov z databázy.

#### 5. Ukážka vstupu a výstupu

Vstup:

```
python main.py "Obžalovaný byl přistižen při neoprávněném vstupu do bytového domu, kde došlo k poškození společného majetku. Dále se zjistilo, že měl u sebe nástroje, které lze považovat za přípravu k vloupání, a je podezřelý z účasti na dalších obdobných incidentech v okolí. Při výslechu uvedl, že šlo o první incident a že neměl v úmyslu někomu ublížit."
```

Výstup:

Odporúčané kroky:

1. Konzultace s advokátem specializovaným na trestní právo, zejména v oblasti vloupání a poškození cizí věci.
2. Přezkoumání předchozích incidentů obžalovaného, aby bylo možné posoudit recidivu a případné zhoršující okolnosti.
3. Zvážení možnosti dohody o vině a trestu s ohledem na závažnost činu a riziko recidivy.
4. Doporučení pro preventivní opatření v bytovém domě (např. zlepšení zabezpečení).

Predikce rozhodnutí soudu:

- Pravděpodobně podmíněný trest s dohledem, možnost pokuty a náhrady škody.
- V případě zjištění dalších incidentů může být trest přísnější.

Použité případy: ID 5, ID 11, ID 19

## Štruktúra programu a hlavné funkcie

Program je rozdelený do niekoľkých častí:

### 1. Konfigurácia

- Definovanie konštánt a zoznamu slov, ktoré sa ignorujú pri porovnávaní textu.

### 2. Funkcie

- `load_cases(filename)`: Načíta prípady zo súboru cases.txt a uloží ich ako zoznam slovníkov. Každý slovník obsahuje kľúče ako ID, Popis, Paragrafy, Rozhodnutí, Trest.
- `find_similar_cases(cases, scenario, top_n)`: Porovná nový prípad s databázou a vypočíta skóre podobnosti na základe počtu zhodných slov vo Predmět, Popis a Paragrafy. Vrátí top N najpodobnejších prípadov.
- `create_prompt_for_ai(new_case, similar_cases)`: Vytvorí textový prompt pre OpenAI model, ktorý obsahuje popis nového prípadu a vybrané podobné prípady s ich informáciami.
- `get_ai_response(prompt)`: Odosiela prompt do OpenAI API a vracia odpoveď modelu s odporúčanými krokmi a predikciou rozsudku.

### 3. Main

- Načíta prípady zo súboru.
- Spojí vstup z príkazového riadku do jedného popisu prípadu.
- Vyhľadá top N podobných prípadov.
- Vytvorí prompt a zavolá OpenAI API.
- Výsledok vypíše do command-line.

## Záver

Dúfam, že tento projekt poskytuje prehľad o mojej práci a schopnostiach pri tvorbe programu kombinujúceho analýzu textu a umelú inteligenciu. Ďakujem za možnosť skúsiť si takýto projekt a prezentovať jeho funkcionality. Verím, že program je zrozumiteľný a demonštruje zamýšľanú funkcionality.