Zdravotnictví v ČR

Technická zpráva k projektu do předmětu UPA
FIT VUT v Brně, 2021

Název týmu
Tým xloren15

Autoři
Jan Lorenc, Bc. (xloren15)
Marek Hlavačka, Bc. (xhlava50)
Martin Smetana, Ing. (xsmeta10)

Specifikace projektu

V rámci projektu je zapotřebí stáhnout datové sady o poskytovatelích zdravotnických služeb na území ČR a o populaci. Tyto je dále třeba analyzovat a pročistit. Na závěr se upravená data musí uložit do NoSQL databáze.

Projekt lze vnímat jako tři části. První je návrh a implementace získání dat z daného zdroje. Další je jejich analýza, úprava a čištění. Sem spadá kontrola typů, chybějících polí, neplatných hodnot nebo úvaha nad možnostmi propojení datových kolekcí. Poslední částí je pak uložení dat do databáze.

Pro kvalitní práci s datovými sadami a zkušenosti členů týmu byl pro implementaci zvolen jazyk Python a pro manipulaci s daty je použita knihovna Pandas. Datové sady se poté ukládají ve formě kolekcí do MongoDB.

Stažení datových sad

Při získávání dat z internetu je dobré si uvědomit, že přestože požadavek na nějaké url typicky získá cílový obsah, při opakovaném stahování existuje šance na to být považován za robota. Cílový web nemusí být ochranou proti tzv. botům vybaven, ale pokud ano, tak skripty pro automatické stahování nemusí být úspěšné. Z toho důvodu je třeba dodat do webových dotazů hlavičky, které z nich udělají platné a důvěryhodné požadavky. Za parametry hlavičky stačí použít jen nějaké základní, jakými jsou například accept, content-type nebo user-agent.

Získání dat v projektu zařizuje třída *DataDownloader* v souboru */src/downloader.py*. Ta ve svém konstruktoru definuje hlavičku pro dotazy a url adresy s datovými sadami.

V metodě *get_healthcare_providers_data()* třída stahuje csv soubory s informacemi o poskytovatelích zdravotnických služeb v ČR. Těchto je na cílové stránce několik a každý měsíc přibývá nový. Z toho důvodu nelze posílat požadavky na stažení konkrétních souborů, neboť by se musely udržovat a pravidelně aktualizovat desítky adres. Lepším řešením je nalezení HTML elementu obsahujícím odkazy a požadavky provést na všechny odkazy uvnitř. Tak se stáhnou všechny aktuálně dostupné datové sady. Požadavky jsou prováděny pomocí modulu *requests* a práci s DOM stromem zařizuje *BeautifulSoup* z modulu *bs4*. Metoda vrací slovník, kde klíče jsou názvy datových sad (datumy, k nimž jsou aktuální) a hodnoty pak obsah stažených csv souborů jako Pandas DataFrame.

Data o populaci jsou již na zdrojovém webu umístěny pouze v jednom pevném souboru. Zde lze proto požadavek provést přímo na jeho url, což dělá metoda *get_population_data()*. Návratovou hodnotou je Pandas DataFrame obsahující obsah souboru o populaci z daného datového zdroje.

Obě metody obsahují volitelný parametr *save_data*, který určuje, jestli se mají stažené soubory uložit lokálně. Toto samo o sobě není pro výsledné řešení žádoucí, proto je to volitelné, nicméně pomáhá to při vývoji. Data si tak lze prohlédnout a analyzovat. Navíc to urychluje následnou práci, jelikož není nutné dlouho čekat na stažení všech dat při každém debug spuštění během čištění dat a práce s databází. Data se při jejich lokální existenci natáhnout ze stažených souborů.

Analýza a úprava dat

Úpravu dat má na starosti třída *DataChecker* nacházející se v souboru *src/data_checker.py*. V konstruktoru třídy dochází k inicializace slovníků sloužících k přetypování, přejmenování hlaviček sloupců a kontrole validity hodnot.

Kontrola dat Českého statistického úřadu o obyvatelstvu ČR probíhá metodou *check_population(df)*.

- *df* – vstupní data ve formátu Pandas DataFrame.

V první fázi dochází k nahrazení hodnot NaN a NaT číslem -1, které nám značí chybějící hodnoty. Následně se data převedou do Numpy pole nad jímž se provádí úpravy dat. Byla-li zaznamenána nekorektní hodnota dat, dojde k jejich nahrazení číslem -2. Po kontrole dochází k převedení dat zpět do datového rámce. Na závěr jsou jednotlivé sloupce přejmenovány a přetypovány.

Kontrolu dat Poskytovatelů zdravotní péče zajišťuje metoda *check_providers(df)*

- *df* – vstupní data ve formátu Pandas DataFrame.

V první fázi stejně jako při kontrole dat o obyvatelstvu dochází k nahrazení NaN a NaT hodnot číslem -1 a následný převod do Numpy pole.
Při provádění úprav je nutné kontrolovat počet sloupců, neboť starší datové sady obsahují jinak strukturovaná data. Jsou to soubory s datem starším než **1.4.2020**, které obsahují pouze 38 sloupců. Poté se přidaly 3 sloupce (KodZZ, DruhZarizeniKod a DruhZarizeniSekundarni) a od **1.7.2020** byl přidán ještě 1 sloupec (DruhPoskytovatele). Následně je již struktura zachována až do **1.11.2021**, kdy dochází k přidání 3 nových hodnot (RozsahPece, PocetLuzek a PoznamkaKeSluzbe).

V rámci dat se kontrolují validní jména a kódy krajů, jména a kódy okresů, či kladná hodnota pořadové číslo zařízení a pořadové číslo detašovaného pracoviště.

Na závěr dochází zpět k převodu na datový rámce, přejmenování a přetypování jednotlivých sloupců. Zde bylo taktéž potřeba kontrolovat počet sloupců, tak aby došlo ke správnému přejmenování a přetypování.

Data nebyla nijak slučována, aby byl zachován jejich vývoj v čase. U poskytovatelů zdravotní péče není problém získat jakákoliv data napříč sadami díky stejným názvům sloupců. Populaci pak lze na poskytovatele napojit skrze území. Data nabízí i napojení na externí zdroje. Například díky GPS souřadnicím by bylo možné poskytovatele zobrazit na mapě.

Výsledkem jsou datové sady se zkontrolovanými hodnotami, jednotně vyřešenými chybějícími či neplatnými hodnotami a jednotnými datovými typy a názvy sloupců.

Uložení do databáze

Data jsou po jednotlivých Pandas datových rámcích ukládána do kolekcí. Data o poskytovatelích zdravotních služeb jsou ukládána s názvem *providers_datum*, zatímco data o populaci jsou uložena v kolekci s názvem *population*.

Pro práci s databází byla zvolena knihovna *PyMongo*, která nám umožňuje práci s MongoDB prostřednictvím jazyka Python. Ukládání dat zajišťuje třída *DataConvertor* v souboru *src/data_convertor.py*, která se ve svém konstruktoru připojuje k databázi a následně připojení otestuje. Nepovede-li se připojit k databázi, program je ukončen.

Převod dat z Pandas datového rámce do databáze probíhá metodou *df_to_mongodb(df, collection_name, drop)*.

- *df* – vstupní data ve formátu Pandas DataFrame,
- *collection_name* – název kolekce pro uložení dat,
- *drop* – parametr nabývající hodnot True/False. Sloužící pro vymazání již existující kolekce se stejným názvem.

Nejprve dochází k vytvoření/načtení kolekce do které se uloží datový rámec. Před uložením se restartuje index a následně jsou data převedena na tvar [{sloupec -> hodnota}, … , {sloupec -> hodnota}]. V tomto tvaru se provede vložení do databáze metodou *insert_many()*.

Pro následný převod dat z databáze zpět do Pandas datového rámce slouží metoda *mongodb_to_df(collection_name, query, no_index, aggr)*.

- *collection_name* - název kolekce
- *query* - dotaz pro metodu *find()*
- *no_index* - parametr pro vymazání sloupce s *_id*
- *aggr* - agregační funkce pro metodu *aggr()*

Na základě parametrů dojde výběru dat se zadané kolekce podle požadavků a následnému převodu do datového rámce, který je funkcí navrácen.

Dále třída obsahuje pomocné metody *get_collection_names()* pro výpis všech kolekcí v databázi a *mongodb_drop(collection_name)* k mazání kolekcí.

Spuštění řešení

Řešení běží na technologiích MongoDb a Python. Z toho důvodu je prerekvizitou ke spuštění nainstalovaný Python verze 3.8. V rámci něj se dále využívají následující moduly, které taktéž vyžadují instalaci: Numpy, Pandas, Requests, Bs4, Pymongo. Databáze MongoDb je spouštěna lokálně přes docker-compose, tedy další a již poslední prerekvizitou je nainstalovaný docker.

Struktura projektu je taková, že ve složce */src* se nacházejí moduly obstarávající potřebnou funkcionalitu. Skript *run.py* poté řešení spustí a s využitím těchto modulů data stáhne, zpracuje a uloží do databáze. Významným souborem je i *docker-compose.yml*, jenž obsahuje konfiguraci MongoDb pro docker.

Za předpokladu nainstalovaných výše zmíněných technologií se řešení spouští následovně:

1. Spustit docker engine (na Windows např. pomocí Docker Desktop)
2. V kořenovém adresáři řešení provést následující příkazy:
   `docker-compose up` … spustí MongoDb v dockeru
   `python3 ./run.py` … spustí řešení projektu