Projekt 2020 - část 1

předmět Zpracování a vizualizace dat v prostředí Python

Changelog

20. 10. 2020: Povoleny další knihovny: requests, BeautifulSoup

Cílem projektu je získat, zpracovat a analyzovat data dostupná na internetu.

Orientační struktura projektu

- Část 1 (20 b)
  - stažení a zpracování dat
  - jednoduchá vizualizace
- Část 2 (20 b)
  - různé pohledy na data (např. dny, kdy se stává nejvíc nehod a podobně)
  - pokročilá vizualizace výsledků
  - zpracování závěrů (porozumění datům)
- Celkový projekt (60 b)
  - znázornění dat na mapě, operace nad těmito daty
  - korelace a predikce
  - automatické vytváření částí zpráv
  - spojení do analytické zprávy

V této části stáhnete volně dostupná data o nehodách v ČR a jejich následcích a budete je zpracovávat. V dalších částech využijete kód z této první části.

Získání a předzpracování dat (20 bodů)

Z URL adresy https://ehw.fit.vutbr.cz/izv/ stáhněte Statistiky nehodovosti Policie ČR pro jednotlivé roky. Seznamte se se strukturou datových položek v dokumentu na stránkách (pozor na duplicitní data v jednotlivých měsících!). Vytvořte jeden souhrnný dataset agregující data přes všechny roky a vybrané kraje. Při zpracování surových dat dejte pozor na časové údaje a neplatné hodnoty (např. minuty rovny 60 znamenají neznámou minutu; hodiny rovny 25 znamenají úplně neznámý čas nehody).

Nad očištěnými vstupními daty vytvořte skript, který vygeneruje statistiku počtu nehod v jednotlivých rocích a jednotlivých krajích (v grafický výstup).

Ve všech dalších úkolech budete pracovat s tímto předzpracovaným souborem *(pokud byla jeho implementace chybná, bude možné ji nahradit; chyba v prvním úkolu samozřejmě neznamená neúspěch v dalších částech).*

Technické řešení

Vytvořte soubor `download.py`, který bude obsahovat třídu `DataDownloader`. Tato třída bude implementovat následující metody (můžete samozřejmě přidat další vlastní, případně můžete přidat další parametry do funkcí, pokud to uznáte za vhodné):

- `__init__(self, url="https://ehw.fit.vutbr.cz/izv/", folder="data", cache_filename="data_{}.pkl.gz")`
  inicializátor - obsahuje volitelné parametry:
  - `url` - ukazuje, z jaké adresy se data načítají. Defaultně bude nastavený na výše uvedenou URL.
  - `folder` - říká, kam se mají dočasná data ukládat. Tato složka nemusí na začátku existovat!
  - `cache_filename` - jméno souboru ve specifikované složce, které říká, kam se soubor s již zpracovanými daty z funkce `get_list` bude ukládat a odkud se budou data brát pro další zpracování a nebylo nutné neustále stahovat data přímo z webu. Složené závorky (formátovací řetězec) bude nahrazený tříznakovým kódem (viz tabulka níže) příslušného kraje. Pro jednoduchost podporujte pouze formát "pickle" s kompresí gzip.
- `download_data(self)`
  funkce stáhne do datové složky `folder` všechny soubory s daty z adresy `url`.
- `parse_region_data(self, region)`
  pokud nejsou data pro daný kraj stažená, stáhne je do datové složky `folder`. Poté je pro daný region specifikovaný tříznakovým kódem (viz tabulka níže) **vždy** vyparsuje do následujícího formátu dvojice (`tuple`), kde první položka je seznam (`list`) řetězců a druhá položka bude seznam (`list`) NumPy polí, schematicky:
  `tuple(list[str], list[np.ndarray])`
  Seznam řetězců odpovídá názvům jednotlivých datových sloupců, NumPy pole budou obsahovat data. Platí, že délka obou seznamů je stejná, `shape` všech NumPy polí je stejný. Při parsování přidejte nový sloupec "region", který bude obsahovat tříznaký kód patřičného kraje, tj. odpovídá hodnotě `region`. Pro každý sloupec zvolte **vhodný datový typ** (t.j. snažte se vyhnout textovým řetězcům, vyřešte desetinnou čárku atp.).
- `get_list(self, regions = None)`
  Vrací zpracovaná data pro vybrané kraje (regiony). Argument `regions` specifikuje seznam (list) požadovaných krajů jejich třípísmennými kódy. Pokud seznam není uveden (je použito None), zpracují se všechny kraje včetně Prahy. Výstupem funkce je dvojice ve stejném formátu, jako návratová hodnota funkce `parse_region_data`.

  Pro každý kraj získá data s využitím funkce `parse_region_data` tak, že se budou výsledky uchovávat v paměti (v nějakém atributu instance třídy) a ukládat do pomocného cache souboru pomocí následujícího schématu:
  - pokud už je výsledek načtený v paměti (tj. dostupný ve vámi zvoleném atributu), vrátí tuto dočasnou kopii
  - pokud není uložený v paměti, ale je již zpracovaný v cache souboru, tak načte výsledek z cache, uloží jej do atributu a vrátí.

    - jinak se zavolá funkce `parse_region_data, výsledek volání se` uloží do cache, poté do paměti a výsledek vrátí

Pokud bude skript spuštěný jako hlavní (t.j. nebude importovaný jako modul)[1], stáhněte data pro 3 vámi vybrané kraje (s využitím funkce `get_list`) a vypište do konzole základní informace o stažených datech (jaké jsou sloupce, počet záznamů a jaké kraje jsou v datasetu).

Dále vytvořte soubor `get_stat.py`, který bude obsahovat funkci `plot_stat(data_source, fig_location = None, show_figure = False)` která vezme zpracovaná data z vašeho `DataDownloader` (při volání použijete `data_source = DataDownloader().get_list([vase_vybrane_kraje])`), načte data a vizualizuje počty nehod.

- Pro každý rok bude jeden samostatný podgraf.
- Jednotlivé grafy budou sloupcové, kdy na ose x bude právě zvolený kraj.
- Sloupce budou anotované pořadím počtu nehod v daném kraji (t.j. kraj s největším počtem nehod bude mít nad patřičným sloupcem 1., atd).
- Graf by měl splňovat všechny náležitosti, které u grafu očekáváme, měl by být přehledný a jeho velikost by měla být taková, aby se dal čitelně použít v šířce A4 (t.j. cca 18 cm). Toto omezení není úplně striktní, ale negenerujte grafy, které by byly přes celý monitor.
- Pokud bude nastavený "`fig_location`", tak se do dané adresy uloží obrázek. Pokud by složka, kam se má obrázek ukládat, neexistovala, vytvořte ji.
- Pokud je nastavený parametr "`show_figure`", tak se graf zobrazí v okně (`plt.show()`).
- Při kreslení používejte přímo knihovnu **`matplotlib`**, ne `seaborn` či funkci `pandas.DataFrame.plot()`!

Pokud bude skript spuštěný jako hlavní (t.j. nebude importovaný jako modul), pomocí modulu argparse povolte volitelné parametry --fig_location a --show_figure a umožněte spuštění funkce z příkazové řádky.
Pozor - tento skript bude využívat vaši implementaci v souboru `download.py`. Aby bylo možné hodnotit tuto podčást, je tedy nutné mít funkční i stahování dat! Pro zpracování dat není na této úrovni dovoleno použít pokročilých knihoven jako je Pandas. Kromě vestavěných knihoven (os, sys, re, requests …) byste si měli vystačit s: gzip, pickle, csv, zipfile, numpy, matplotlib, BeautifulSoup. Další knihovny je možné použít po schválení opravujícím (např ve fóru WIS).

Dokumentace všech částí (souborů, tříd a funkcí) bude přímo v odevzdaných souborech. Snažte se dodržovat konvenci PEP 257 [https://www.python.org/dev/peps/pep-0257] a PEP 8 [https://www.python.org/dev/peps/pep-0008/].

[1] Při spuštění `python3 download.py` (lze testovat pomocí podmínky `__name__=="__main__"`)

Kódy krajů

Ve svém kódu využívejte následující kódy (bez diakritiky, s velkými písmeny) pro identifikaci krajů.

Jméno	Kód dle ČSÚ
Hlavní město Praha	PHA
Středočeský	STC
Jihočeský	JHC
Plzeňský	PLK
Karlovarský	KVK
Ústecký	ULK
Liberecký	LBK
Královéhradecký	HKK
Pardubický	PAK
Olomoucký	OLK
Moravskoslezský	MSK
Jihomoravský	JHM
Zlínský	ZLK
Kraj Vysočina	VYS

Odevzdávání a hodnocení

Do 11. 11. 2020 odevzdejte dva samostatné soubory. Pro práci používejte
Pro práci s projektem je možné využít všechny knihovny představené při přednáškách. Skripty musí všechna pomocná data ukládat do složky “./data/” (viz parametr folder), kterou si případně vytvoří.

Hodnotit se bude zejména:

- správnost výsledků
- vizuální dojem z grafů
- kvalita kódu
    - efektivitu implementace a reprezentace (i rychlost v porovnání s ostatními řešeními)
    - přehlednost kódu
    - dodržení standardů a zvyklostí pro práci s jazykem Python
    - dokumentace funkcí a souborů
    - znovupoužitelnost kódů - správná izolace potřebných částí do funkcí

Celkem za první část můžete získat až 20 bodů, přičemž je k zápočtu nutné získat z této části minimálně 2 body.

Dotazy a připomínky

Na fóru WIS případně na mailu mrazek@fit.vutbr.cz.