Zadanie zaliczeniowe NYPD

Wersja 1.2

Zadaniem zaliczeniowym jest stworzenie biblioteki do przetwarzania pewnego typu danych. Do biblioteki powinien być załączony skrypt (może być w formie *Jupyter notebooka*) pokazujący użycie biblioteki na konkretnych danych. Własne propozycje zadań również muszą to spełniać.

Podstawowa propozycja zadania:

Bazując na danych od dochodach budżetowych pobieranych przez urzędy skarbowe na rzecz jednostek samorządu terytorialnego za rok 2020 policz dochód każdej jednostki samorządu terytorialnego z podatku PIT. Porównaj ten dochód z dochodem za rok 2019.

- https://www.gov.pl/web/finanse/udzialy-za-2020-r
- https://www.gov.pl/web/finanse/udzialy-za-2019-r

Następnie połącz te dane z danymi o ludności w poszczególnych Jednostkach Samorządu Terytorialnego

- https://stat.gov.pl/obszary-tematyczne/ludnosc/ludnosc/ludnosc-stan-i-struktura-ludnosci-oraz-ruch-naturalny-w-przekroju-terytorialnym-stan-w-dniu-31-12-2020,6,29.html

i policz średni dochód opodatkowany dla tych jednostek. Dla Województw i Powiatów policz wariancję dochodu w podległych jednostkach samorządu terytorialnego i porównaj przewidziany dochód dla tej jednostki ze średnią ważoną jednostek podległych.

Dla uproszczenia można przyjąć, że wszyscy są opodatkowani w ramach pierwszego progu. Można założyć arbitralny procent pracujących między 50 a 100%.

Wszystkie ścieżki do plików (i inne parametry) powinny być zgrupowane:

- w jednej z pierwszych komórek *Jupyter notebooka* albo
- podawane w wierszu poleceń i odczytywane przez skrypt za pomocą biblioteki *argparse*.

W bibliotece nie wolno zapisywać literałów tekstowych ze ścieżkami dostępu do plików, a wszystkie ścieżki powinny być podawane jako parametry funkcji.

Jeśli w trakcie analizy danych wykryjesz jakieś niezgodności, to program powinien je zgłaszać, w sposób pozwalający na ocenę istotności niezgodności, natomiast sposób ich obsługi możesz przyjąć według własnego uznania (np. pomijając niezgodne dane).

Częścią rozwiązania jest raport z wynikiem analizy (wynik działania notebooka, wynik uruchomienia skryptu, arkusz kalkulacyjny z zapisanymi wynikami) oraz niezbędnymi informacjami pośrednimi.

Przy ocenianiu pod uwagę będą brane następujące punkty:

- Wyjaśnienia dotyczące kodu i analizy podczas egzaminu, w tym w jaki sposób problemy napotkane w danych wpłynęło na stworzony kod.

- Poprawny podział kodu na skrypt i bibliotekę, a wewnątrz biblioteki na paczki i moduły.
- Jakość kodu, w tym poprawne nazewnictwo funkcji i zmiennych, krótkie funkcje. Jako inspirację można zajrzeć do PEP8 https://www.python.org/dev/peps/pep-0008/ i spróbować użyć narzędzi jak `flake8` lub `pylint`.
- Pokrycie kodu (biblioteki) testami.
- Możliwość zainstalowania biblioteki jako pakietu pythonowego używając `pip install ./path/to/package`. Ta możliwość ma być wprowadzona jako merge (pull) request do głównej gałęzi [temat będzie omówiony w styczniu].
- Użycie biblioteki *numpy* lub *pandas* w miejscach gdzie zwiększa to czytelność kodu.
- Wynik profilowania kodu wraz ze sprawdzeniem, czy są w programie wąskie gardła i sugestią jak je można poprawić.

W związku z powyższymi wymaganiami zadanie zaliczeniowe musi być oddane przez repozytorium Git przynajmniej 24 godziny przed egzaminem.

Potencjalne trudności:

1. Zmiany kodów terytorialnych w związku ze zmianami administracyjnymi.
2. Konieczność samodzielnego sprawdzenia jaki procent PIT należy się poszczególnym jednostkom samorządu.
3. Miasta na prawach powiatu.
4. Należy pamiętać, że podatek PIT płacą też emeryci.

Historia zmian:

1.2 [2021-01-14]: Dodanie akapitu upraszczającego (1 próg podatkowy, 50%-100% pracujących), dodanie akapitu doprecyzowującego oczekiwaną postać wyniku (raport z wynikiem analizy).