Przetwarzanie i analiza dużych zbiorów danych 2020/21

Prowadzący: mgr inż. Rafał Woźniak środa, 11:45

Piotr Wardęcki	234128	234128@edu.p.lodz.pl
Paweł Galewicz	234053	234053@edu.p.lodz.pl
Bartosz Jurczewski	234067	234067@edu.p.lodz.pl

Zadanie 1: Analiza i porównanie czasów wykonywania zapytań przy użyciu różnych narzędzi programistycznych

1. Cel

Celem zadania była implementacja zapytań do zestawu danych dotyczących zgłoszeń na numer 3-1-1 w Nowym Jorku [1] przy użyciu kilku narzędzi programistycznych, porównanie czasów ich wykonania oraz próba optymalizacji zaproponowanych rozwiązań.

2. Wprowadzenie

Zestaw danych zawiera informacje na temat zgłaszanych incydentów, natomiast na potrzeby zadania najistotniejszymi z nich są następujące kolumny:

— Agency Name - nazwa urzędu odpowiedzialnego za rozwiązanie incydentu
— Complaint Type - rodzaj zgłoszonego incydentu
— Borough - dzielnica której dotyczy zgłoszenie

Analizę czasów wykonywania kwerend przeprowadzimy na następujących zagadnieniach:

— Znalezienie najczęściej zgłaszanych skarg
— Znalezienie najczęściej zgłaszanych skarg w każdej dzielnicy
— Znalezienie urzędów, do których najczęściej zgłaszano skargi

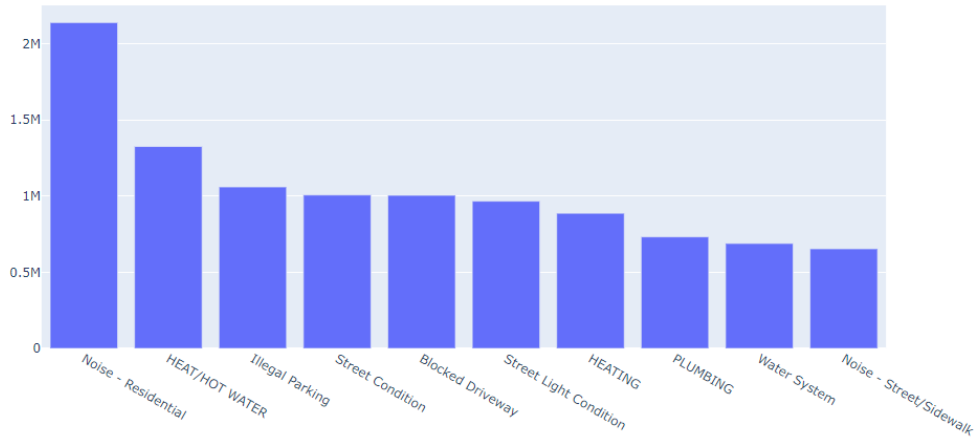
3. Opis implementacji

Algorytmy potrzebne do zadania zostały zaimplementowane za pomocą języka Python w wersji 3.8.2. Wykorzystano w nim biblioteki Pandas, Pyodbc, Sqlalchemy oraz Dask. Dodatkowo do wygenerowania wykresów oraz usprawnienia naszej pracy zdecydowaliśmy się użyć Jupyter Notebook [2].

4. Wyniki kwerend

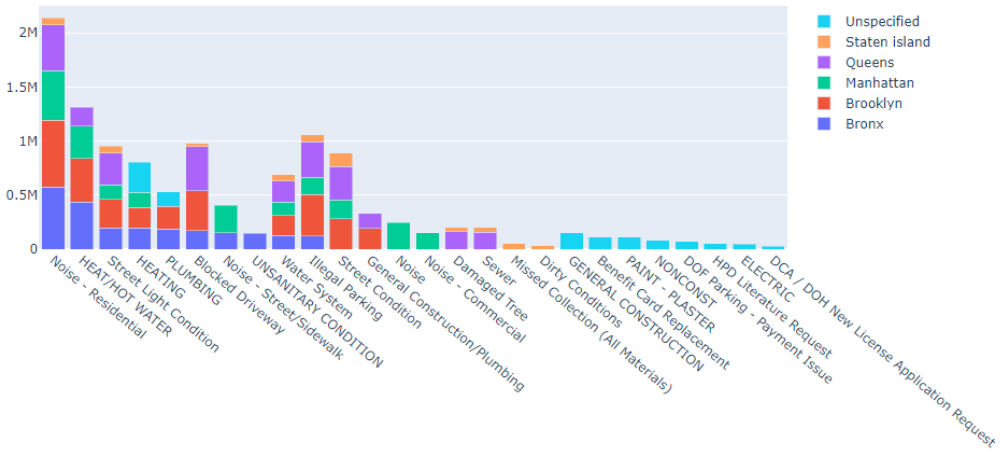
Sekcja ta prezentuje wykresy z wynikami zapytań. Dla każdej technologii wyniki prezentowały się dokładnie tak samo - co jest dowodem na brak logicznych rozbieżności między implementacjami - dlatego wynik każdej kwerendy zaprezentowany został raz. W pliku Notebook dostępne są osobne wykresy dla każdej technologii.

4.1. Najczęściej zgłaszane skargi



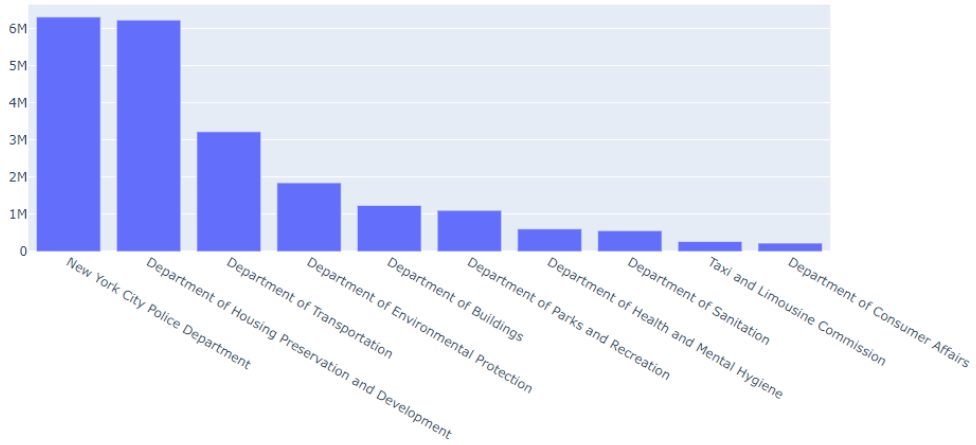
Rysunek 1. Najczęściej zgłaszane skargi

4.2. Najczęściej zgłaszane skargi w każdej dzielnicy



Rysunek 2. Najczęściej zgłaszane skargi w każdej dzielnicy

4.3. Urzędy, do których najczęściej zgłaszano skargi



Rysunek 3. Urzędy, do których najczęściej zgłaszano skargi

5. Wyniki

Po przeprowadzonych badaniach czasów wykonywania operacji otrzymaliśmy następujące wyniki:

Tabela 1. Porównanie czasu wykonywania kwerend

Czynność	Python	SQLite3	MSSQL
Przygotowanie danych	1m 15.78s	3m 30s (1)	7m 16.15s (2)
Najczęściej zgłaszane skargi	1.89s	17.7s	0.529s
Najczęściej zgłaszane skargi w każdej dzielnicy	4.14s	60.66s	2.23s
Urzędy, do których najczęściej zgłaszano skargi	1.75s	19.5s	0.527s

Ad. 1 - Na przygotowanie danych składało się przygotowanie danych oraz stworzenie na ich podstawie bazy plikowej SQLite3

Ad. 2 - Na przygotowanie danych składało się wygenerowanie pliku z danymi, stworzenie i wypełnienie bazy danych

6. Optymalizacja

W celu optymalizacji kwerend wykorzystujących bazy danych postanowiliśmy zastosować mechanizm indeksów. W przypadku danych przechowywanych w MSSQL nie mogliśmy utworzyć indeksów na istniejących wierszach, ponieważ dane przechowywane w tabeli nie były unikatowe. Problem ten

rozwiązaliśmy dodając nową kolumnę *Id* (numer rzędu dla każdej krotki) i na niej utworzyliśmy indeks. Do użycia ich w zapytaniach wymagane było użycie odpowiedniej składni. *SQLite3* automatycznie dodał kolumnę z id oraz utworzył na niej indeks podczas importu danych. Dodatkowo silnik używa ich kiedy to tylko możliwe.

W celu optymalizacji czasu wykonania kwerend zaimplementowanych w języku Python zdecydowaliśmy się użyć biblioteki *Dask*, która dostarcza implementację API modułów Pandas, w tym klasy Dataframe, która ma być przeznaczona do wykorzystywaniu w zagadnieniach BigData i która tworzy obiekty Dataframe z mniejszym wykorzystaniem pamięci operacyjnej w stosunku do biblioteki Pandas.

Tabela 2. Przygotowanie danych.

	Python	**SQLite3**	**MSSQL**
Czas	1m 15.78s	3m 30s	7m 16.15s
Czas po optymalizacji	2.88s	n/d	13m 45.15s

Tabela 3. Najczęściej zgłaszane skargi.

	Python	**SQLite3**	**MSSQL**
Czas	1.89s	17:07s	0.529s
Czas po optymalizacji	36.04s	n/d	6.49s

Tabela 4. Najczęściej zgłaszane skargi w każdej dzielnicy.

	Python	**SQLite3**	**MSSQL**
Czas	4.14s	99.6s	2.23s
Czas po optymalizacji	37.4s	n/d	44s

Tabela 5. Urzędy, do których najczęściej zgłaszano skargi.

	Python	**SQLite3**	**MSSQL**
Czas	1.75s	19.05s	0.527s
Czas po optymalizacji	36.7s	n/d	06.16s

7. Wnioski

— Wykorzystanie narzędzi do BigData przy analizie nie dużych zbiorów danych może przynieść skutki odwrotnie z oczekiwanymi.
— Jedna metoda optymalizacji nie musi sprawdzić się dla każdego zapytania.
— Indeks na pole które nie jest badane zawsze wydłuża czas zapytania.
— Optymalizację należy dostosowywać do używanej technologi oraz narzędzia.

Bibliografia

[1] *311 Service Requests from 2010 to Present* https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

[2] *Jupyter notebook do zadania pierwszego* https://github.com/jurczewski/PiADZD/blob/master/zad1/zad1.ipynb