ANALIZA RUCHU LOTNICZEGO ORAZ

BADANIE POTENCJALNYCH KOLIZJI

Adam Nowak

Uniwersytet Warszawski

Warszawa

25.06.2024

Spis treści

[1. Cel projektu 3](#_Toc170245519)

[2. Przegląd danych oraz obróbka danych 3](#_Toc170245520)

[3. Przygotowanie danych do programu 4](#_Toc170245521)

[4. Główny program 5](#_Toc170245522)

[4.1 Biblioteki 5](#_Toc170245523)

[4.2 Wczytanie danych 5](#_Toc170245524)

[4.3 Funkcje 5](#_Toc170245525)

[4.4 Wywołanie funkcji 8](#_Toc170245526)

[4.5 Wyniki oraz wnioski 8](#_Toc170245527)

[4.6 Wizualizacja wyników 10](#_Toc170245528)

[5. Spis ilustracji 11](#_Toc170245529)

# 1. Cel projektu

Celem projektu była analiza ruchu lotniczego oraz badanie potencjalnych kolizji. Do zadania wykorzystano bazę danych „fsig-raw” dostępną na serwerze fsig-students.ocean.icm.edu.pl.

Program został napisany w języku Python ze względu na dużą ilość danych. Na serwerze użyto wersji 2.7, natomiast po uzyskaniu danych lokalnie wykorzystano wersję 3.11 oraz środowisko Jupyter Notebook.

# 2. Przegląd danych oraz obróbka danych

Początkowo skupiono się na przeanalizowaniu dostępnych danych. Każdy zestaw zawiera kolejno pola: pitr, type, ident, alt, clock, lon, lat oraz inne istotne zmienne (Rysunek 1).

Obraz zawierający tekst, Czcionka, biały, dokument

Opis wygenerowany automatycznie

Rysunek 1 Przegląd danych

Z zestawu o parametrze type: flight plan początkowo wyciągnięto identyfikatory lotów (ident) oraz rejestracje samolotu (reg) (Rysunek 2). Otrzymano w ten sposób plik typu JSON, w którym zapisano kolejne rekordy z danymi (Rysunek 3).

Obraz zawierający tekst, Czcionka, linia, Wykres

Opis wygenerowany automatycznie

Rysunek 2 Wycinek kodu, kryteria wyszukiwania

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek 3 Dane zawierające identyfikator lotu

Po otrzymaniu identyfikatorów lotu zdecydowano się wyszukać 15 lotów o otrzymanych nazwach, lecz z datą 02-01-2018. Loty 1 stycznia często nie posiadały punktów startowych, ponieważ od początku znajdowały się w powietrzu. Dla znalezionych identyfikatorów lotu wyciągnięto dane z typu „Position”, takie jak: identyfikator, numer rejestracyjny samolotu, długość geograficzną, szerokość geograficzną oraz wysokość. Czas PITR w tym przypadku nie był potrzebny ze względu na czas clock, który pozwalał określić czas danego lotu. Parametr air\_ground, określający, czy samolot jest w powietrzu lub na ziemi, również nie był potrzebny ze względu na posiadaną wysokość (Rysunek 4).

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek 4 Pobranie niezbędnych danych oraz ich obróbka

W ten sposób otrzymano 15 plików JSON z lotami zawierających niezbędne informacje.

# 3. Przygotowanie danych do programu

Przed rozpoczęciem analizy należało zmodyfikować format danych w plikach i zmergować je w jeden plik. W tym celu napisano krótki program, który najpierw ujednolicał zapis w plikach, a następnie scalał je w jeden plik (Rysunek 5).

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 5 Dalsza obróbka oraz mergowanie danych

# 4. Główny program

Program składa się z części wczytania bibliotek, załadowania danych, trzech funkcji, wywołania funkcji oraz przedstawienia wyników zarówno graficznie, jak i tekstowo.

## 4.1 Biblioteki

* JSON - praca z danymi w formacie JSON (wczytywanie i zapisywanie)
* OS - operacje na plikach i ścieżkach (sprawdzanie istnienia plików)
* GEOPY - obliczenia odległości na powierzchni Ziemi między dwoma punktami
* SHAPEL - reprezentowanie i manipulacja punktami geograficznymi
* DATETIME - operacje na datach i czasach (parsowanie, formatowanie)
* FOLIUM - tworzenie interaktywnych map z zaznaczonymi punktami i liniami
* ITERTOOLS - tworzenie cyklicznych iteratorów
* MATH – obliczenia matematyczne

## 4.2 Wczytanie danych

Dane zostały wczytane z poprzednio utworzonego pliku zawierającego zestawy lotów.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 6 Użyte biblioteki oraz wczytanie danych

## 4.3 Funkcje

1. **check\_collisions(paths, idents)**: Funkcja ta sprawdza kolizje między trasami lotów na podstawie ich ścieżek reprezentowanych przez obiekty LineString. Iteruje po parach ścieżek, sprawdza ich przecięcia i znajduje najbliższe punkty przecięcia. Następnie używa funkcji **find\_altitude\_and\_clock** do pobrania danych wysokościowych, czasowych oraz współrzędnych dla tych najbliższych (Rysunek 9).
2. **find\_flight\_data(ident)**: Funkcja ta służy do wyszukiwania danych lotu na podstawie identyfikatora. Przeszukuje ona dane lotów, aby znaleźć odpowiedni rekord na podstawie identyfikatora. Zwraca rejestrację lotu, wysokość pierwszego punktu docelowego oraz czas pierwszego punktu docelowego, jeśli dostępne są w danych(Rysunek 8Rysunek 8).
3. **find\_altitude\_and\_clock(closest\_point, ident, data)**: Funkcja ta znajduje punkt docelowy lotu o najbliższej odległości do podanego punktu. Iteruje po wszystkich punktach docelowych dla danego lotu, oblicza odległość od danego punktu za pomocą geometrii Shapely i zwraca wysokość, czas, oraz współrzędne geograficzne tego najbliższego punktu (Równanie: 1), (Rysunek 7).

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 7 Funkcja: find\_altitude\_and\_clock

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 8 Funkcja: find\_flight\_data

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 9 Główna funkcja: check\_collisions

## 4.4 Wywołanie funkcji

Następnie, dla każdego lotu z danych, utworzono listy paths i idents, które zostały wypełnione trasami lotów oraz identyfikatorami tych lotów na podstawie punktów przejścia. Kolejno wywołano funkcję check\_collisions(paths, idents), której wyniki, potencjalne kolizje tras lotów, zostały zapisane w liście collisions. Na zakończenie wyniki zostały wyświetlone na ekranie, przedstawiając informacje o potencjalnych kolizjach tras lotów oraz różnice w wysokości i czasie między nimi, jeśli takie zostały znalezione (Rysunek 10).

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 10 Wywołanie funkcji

## 4.5 Wyniki oraz wnioski

Na podstawie wyników zidentyfikowanych kolizji tras lotów (Rysunek 11) można wyciągnąć następujące wnioski:

* W większości przypadków znaleziono loty, których trajektorie przecinały się, jednak różnice w czasie przelotu oraz wysokości były znaczne, co sugeruje dobrze zaplanowane i bezpieczne operacje lotnicze.

Wyjątki stanowią konkretne przypadki:

* Flight SWA622 (Reg: N8633A) and Flight AAL2254 (Reg: N177AN)
* Flight CBJ480 (Reg: B8550) and Flight QFA79 (Reg: VHQPI)
* Flight KAL124 (Reg: HL7586) and Flight QFA79 (Reg: VHQPI)
* Flight QFA79 (Reg: VHQPI) and Flight VOZ214 (Reg: VHYVC)
* Flight AAL2254 (Reg: N177AN) and Flight SWA622 (Reg: N8633A)

W pierwszych trzech przypadkach odnotowaną mała różnicę w pułapie, a samoloty minęły się w odstępie około dwóch godzin. W pozostałych dwóch przypadkach samoloty również minęły się w okolicy dwóch godzin, lecz znajdowały się w tych miejscach na różniących się wysokościach.

Jednakże, takie różnice w czasie przelotu są na tyle duże, że nie można tego traktować jako bezpośrednie zagrożenie.

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Rysunek 11 Wyniki - potencjalne kolizje

## 4.6 Wizualizacja wyników

Wynik graficzny (Rysunek 12) został załączony jako interaktywny plik o nazwie:

*„flight\_route\_map\_with\_collisions.html”.*

*Obraz zawierający mapa

Opis wygenerowany automatycznie*

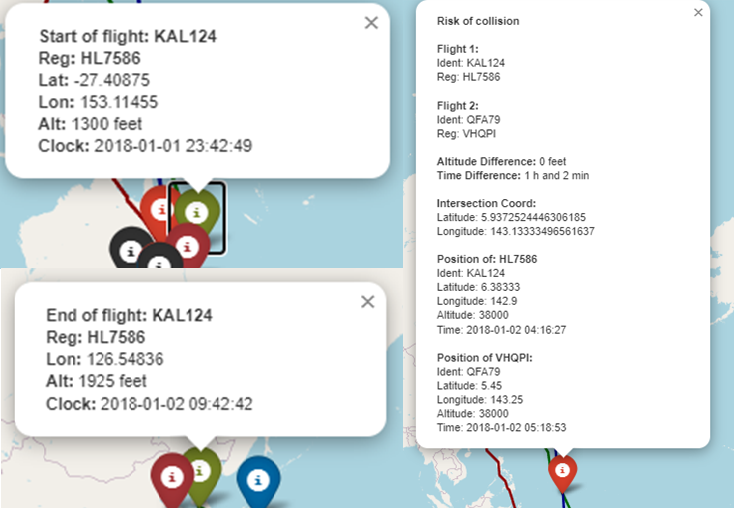
Rysunek 12 Wizualizacja wyników

Objaśnienia (Rysunek 13):

Każda trasa posiada znacznik początkowy oraz końcowy a także ścieżkę oznaczoną tym samym kolorem co markery.

Punkty kolizji są oznaczone kolorem CZERWONYM.

Po najechaniu na danych znacznik zostaną wyświetlone informacje:



Rysunek 13 Objaśnienia

# 5. Spis ilustracji

[Rysunek 1 Przegląd danych 3](#_Toc170245425)

[Rysunek 2 Wycinek kodu, kryteria wyszukiwania 3](#_Toc170245426)

[Rysunek 3 Dane zawierające identyfikator lotu 3](#_Toc170245427)

[Rysunek 4 Pobranie niezbędnych danych oraz ich obróbka 4](#_Toc170245428)

[Rysunek 5 Dalsza obróbka oraz mergowanie danych 4](#_Toc170245429)

[Rysunek 6 Użyte biblioteki oraz wczytanie danych 5](#_Toc170245430)

[Rysunek 7 Funkcja: find\_altitude\_and\_clock 6](#_Toc170245431)

[Rysunek 8 Funkcja: find\_flight\_data 6](#_Toc170245432)

[Rysunek 9 Główna funkcja: check\_collisions 7](#_Toc170245433)

[Rysunek 10 Wywołanie funkcji 8](#_Toc170245434)

[Rysunek 11 Wyniki - potencjalne kolizje 9](#_Toc170245435)

[Rysunek 12 Wizualizacja wyników 10](#_Toc170245436)

[Rysunek 13 Objaśnienia 10](#_Toc170245437)