|  |  |  |
| --- | --- | --- |
|  | **Софийски университет „Св. Кл. Охридски”**  Факултет по математика и информатика  *ИЗВЛИЧАНЕ НА ИНФОРМАЦИЯ И ОТКРИВАНЕ НА ЗНАНИЯ* |  |

**Размити множества**

***Зимен семестър, 2023/2023 год.***

**„Изчисляване на орбита и шанс за колизия на два сателита чрез размити множества“**

**Курсов проект**

*Автори:*

*Мартин Тодоров, 7MI3400322*

Съдържание

[1 Въведение 2](#_Toc1041766259)

[2 Анализ на решението 3](#_Toc664988844)

[2.1 Потребителски изисквания и работен процес 3](#_Toc123109475)

[2.2 Примерен графичен интерфейс 4](#_Toc790123179)

[2.3 Диаграми на дейността 4](#_Toc524875537)

[2.4 Модел на съдържанието/данните 5](#_Toc2104269727)

[3 Алгоритми 6](#_Toc175360141)

[3.1.1 Изчисление на средна височина (Mean Altitude Calculation) 6](#_Toc984072957)

[3.1.2 Оценка на маневреност (Maneuverability Evaluation) 6](#_Toc2118836264)

[3.1.3 Изчисление на орбитално сходство (Orbital Similarity Calculation) 6](#_Toc218791024)

[3.1.4 Определяне на категория риск (Risk Category Determination) 6](#_Toc1957674127)

[Функции за Принадлежност на Риск 7](#_Toc1466461879)

[Функции за Принадлежност на Разстояние 7](#_Toc1815720657)

[Функции за Принадлежност на Относителна Скорост 7](#_Toc1210754627)

[Функции за Принадлежност на Орбитално Сходство 8](#_Toc1161468947)

[Функции за Принадлежност на Маневреност 8](#_Toc1875647086)

[Правило 1: Ако разстоянието е много близко, рискът е много висок. 8](#_Toc1416633811)

[Правило 2: Ако разстоянието е близко, рискът е висок. 8](#_Toc1883061233)

[Правило 3: Ако разстоянието е умерено, рискът е среден. 8](#_Toc720932312)

[Правило 4: Ако разстоянието е далеч, рискът е нисък. 8](#_Toc27486469)

[Правило 5: Ако относителната скорост е бърза И разстоянието е близко И (орбиталното сходство е подобно ИЛИ маневреността е ниска), тогава рискът е много висок. 8](#_Toc935902783)

[Правило 6: Ако разстоянието е близко И относителната скорост е бърза, рискът е висок. 8](#_Toc362697098)

[Правило 7: Ако маневреността е висока И разстоянието е умерено И относителната скорост е умерена, рискът е нисък. 8](#_Toc1608736829)

[4 Тестово изпълнение 9](#_Toc644455737)

[5 Заключение и възможно бъдещо развитие 10](#_Toc403790867)

[6 Използвани технологии и библютеки 10](#_Toc1455565040)

[7 Използвани литературни източници и Уеб сайтове 11](#_Toc2134088229)

# Въведение

Този проект имплементира система за оценка на риска от сблъсъци между сателити, използвайки различни метрики като разстояние, относителна скорост, орбитално приликание и маневреност. Основната идея на проекта е, че резултата се представя под формата на “колко процента увереност имаме” че шанса за сблъсък на два сателита е “висока”.

С разширяването на космическата дейност и увеличаването на броя на сателитите, управлението на орбитален трафик и предотвратяването на сблъсъци стават все по-важни. Традиционните методи за оценка на риска често се базират на прости геометрични модели и детерминистични изчисления, които могат да не отразяват адекватно несигурностите и динамичния характер на орбиталните среди. В този контекст, въвеждането на размитата логика предлага гъвкав подход за моделиране на несигурност и имплицитна сложност, позволявайки по-точно оценяване на риска от сблъсък.

# Анализ на решението

## Потребителски изисквания и работен процес

#### Входни Данни

* **Входно Съдържание:** Потребителят въвежда NORAD идентификаторите на два сателита и дата (в формат YYYY-MM-DD), които представляват входните данни за приложението.
* **Източник на Данни:** Данните се получават директно от потребителя чрез графичния потребителски интерфейс (GUI), създаден с помощта на библиотеката **tkinter**. TLE(Two-line element set)\* данните се зареждат с помоща на norad идентификатора от API-a на celestrak[1].

#### Обработка

* **Валидация на Входни Данни:** Приложението първо проверява дали всички полета са попълнени правилно и дали датата е въведена в правилния формат.
* **Изчисления:**
  + Зарежда Two-Line Element Set (TLE) данни за сателитите чрез библиотеката **skyfield**.
  + Изчислява средната височина, маневреност, и орбитално сходство на сателитите.
  + Използва размита логика (чрез **skfuzzy**) за оценка на риска от сблъсък на база разстояние, относителна скорост, орбитално сходство и маневреност.

\* TLE (Two-Line Element Set) е формат, използван за описване на орбитата на земни сателити. TLE съдържа два реда текстова информация, която представя прецизни математически данни за положението и движението на сателит в орбита около Земята. Форматът е разработен от NORAD (North American Aerospace Defense Command) и се използва широко за проследяване на сателити и космически отпадъци.

#### Изход

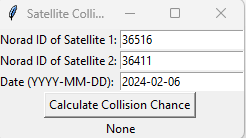
* **Резултати:** Като изход се получава файл в CSV формат, съдържащ подробни резултати от анализа на риска от сблъсък, включително време на проверката, категория на
* риска, процентна стойност, разстояние между сателитите, относителна скорост, орбитално сходство и средно маневреност.
* **Употреба на Резултатите:** Резултатите могат да се използват от операторите на сателити за оценка на потенциалния риск от сблъсък и вземане на решения относно необходимостта от корекции на орбитата, за да се избегне сблъсък. Файлът в CSV формат улеснява анализа и архивирането на данните.

Този процес обхваща целия жизнен цикъл от събирането на входни данни от потребителя, през обработката на тези данни с помощта на астрономически и математически алгоритми и размита логика, до генерирането на подробен отчет, който помага в прогнозирането и управлението на риска от сблъсъци в орбита.

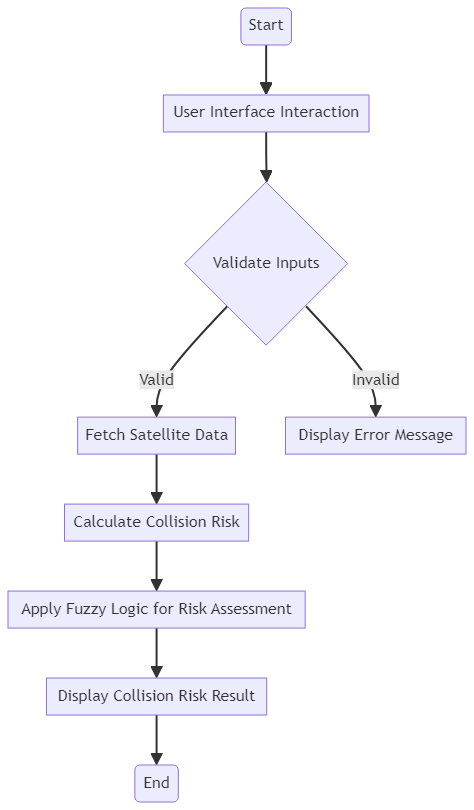
## Примерен графичен интерфейс

Потребителсният интерфейс се съдържа от 3 полета за попълване:

* Идентификатор на Сателит 1.
* Идентификатор на Сателит 2
* Дата за която искам да проверим дали е възоцна колизия
* Бутон за стартиране на изчисленията



## Диаграми на дейността



## Модел на съдържанието/данните

**Описание:** Включва входни данни като NORAD идентификатори на сателитите и датата за анализ. Резултатите от изчисленията, като средна височина, маневреност, риск категории, и процентна стойност на риска също са текстови.

**Размер и Тип:** Малък до умерен размер; данните се записват в .csv файл.

**Кодиране:** Текстовите данни се кодират в UTF-8.

Двата реда на TLE взети чрез използване на Norad идентификатора съдържат следните основни данни:

* **Ред 1:**
  + Номер на каталога на сателита (NORAD Catalog Number)
  + Класификационен статус (секретност)
  + Дата и време на старта
  + Елементи на орбитата (например, епоха на орбитата)
* **Ред 2:**
  + Наклон (Inclination)
  + Дължина на възходящия възел (Right Ascension of the Ascending Node, RAAN)
  + Ексцентричност (Eccentricity)
  + Аргумент на перигея (Argument of Perigee)
  + Средна аномалия (Mean Anomaly)
  + Средно движение (Mean Motion)
  + Прогнозен брой обиколки на ден

# Алгоритми

Този код използва няколко ключови алгоритми и техники за изчисление, анализ и визуализация на данни за орбитални обекти и риск от сблъсък. Ето обобщение на основните алгоритми и методи, използвани в кода:

### **Изчисление на средна височина (Mean Altitude Calculation)**

Алгоритъмът за изчисление на средната височина използва орбиталните параметри на сателита, по-специално неговото средно движение (mean motion), за да определи средната височина на сателита над повърхността на Земята. Това изчисление използва уравнението на Кеплер за орбитален период и гравитационния параметър на Земята.

### **Оценка на маневреност (Maneuverability Evaluation)**

Маневреността се изчислява като функция от средната височина, ексцентричността и наклона на орбитата. Висшата оценка се присъжда на сателити, които се намират на по-високи орбити, поради по-малката им възможност за маневриране поради по-голямата енергия, необходима за промяна на траекторията.

### **Изчисление на орбитално сходство (Orbital Similarity Calculation)**

Този алгоритъм сравнява ключови орбитални елементи на два сателита, като наклон, дължина на възходящия възел, ексцентричност и аргумент на перигея, за да определи тяхното орбитално сходство. Сходството се изразява като числена стойност между 0 и 1, където 1 означава пълно съвпадение на орбитите.

### **Определяне на категория риск (Risk Category Determination)**

Използва се размита логика чрез библиотеката **skfuzzy** за определяне на категорията на риск от сблъсък, като се вземат предвид различни параметри като разстояние между сателитите, тяхната относителна скорост и орбитално сходство. Размита логика позволява по-гъвкаво вземане на решения в условия на несигурност.

#### Входни параметри

* **distance (Разстояние)**: Това е входна променлива , която представлява разстоянието между два обекта в километри. Дефинирана е в обхвата от 0 до 10000 км със стъпка от 1 км. Разстоянието е критичен параметър за оценка на потенциалния риск от сблъсък.
* **risk (Риск)**: Това е изходна променлива, която представлява оценката на риск от сблъсък, изразена като процент от 0 до 100. Категориите на риск са дефинирани чрез триъгълни функции на принадлежност за "много висок", "висок", "среден" и "нисък" риск.
* **relative\_velocity (Относителна Скорост)**: Входна променлива, показваща относителната скорост между два сателита в км/сек. Обхватът е от 0 до 15000 км/сек със стъпка от 1 км/сек. Относителната скорост е важен фактор, който влияе на риска и възможностите за избягване на сблъсък.
* **orbital\_similarity (Орбитално Сходство)**: Входна променлива, която оценява степента на сходство между орбитите на двата сателита, като се изразява в числова стойност между 0 и 1. Това е ключов аспект при оценката на дългосрочния риск от сблъсък.
* **maneuverability (Маневреност)**: Входна променлива, оценяваща способността на сателита да извършва маневри за избягване на сблъсък. Обхватът е от 0 до 100, като по-високите стойности предполагат по-добра маневреност.

#### **Функции на Принадлежност**

Функциите на принадлежност определят как всяка стойност на променливите се картографира към степен на "истинност" или принадлежност към определени множества. За всяка категория са дефинирани триъгълни функции (**fuzz.trimf**), които са характеризирани с три точки, представляващи началото, върха и края на триъгълника. Например, за **risk** категориите са дефинирани така:

### **Функции за Принадлежност на Риск**

* **very\_high**: Дефинира се с триъгълна функция, където рискът се счита за "много висок" при стойности от 0 до 25. Това означава, че ако изчисленият риск е между 0 и 25, степента на принадлежност към категорията "много висок" ще нараства линейно.
* **high**: Рискът е "висок" при стойности между 0 и 50, като върхът на триъгълника е при 25. Това позволява плавен преход между "много висок" и "среден" риск.
* **medium**: Среден риск е при стойности между 25 и 75, с връх на триъгълника при 50.
* **low**: Нисък риск е при стойности между 50 и 100, с връх на триъгълника при 75.

### **Функции за Принадлежност на Разстояние**

* **very\_close**: Разстоянието се счита за "много близко" при стойности от 0 до 150 км.
* **close**: "Близко" при стойности между 100 и 500 км.
* **moderate**: "Умерено" при стойности между 400 и 3000 км.
* **far**: "Далеч" при стойности между 2500 и 10000 км.

### **Функции за Принадлежност на Относителна Скорост**

* **slow**: Относителна скорост е "бавна" при стойности до 2 км/с.
* **moderate**: "Умерена" скорост при стойности между 1 и 7 км/с.
* **fast**: "Бърза" скорост при стойности между 5 и 15 км/с.

### **Функции за Принадлежност на Орбитално Сходство**

* **different**: Орбитално сходство се счита за "различно" при стойности от 0 до 0.5.
* **similar**: Сходство е "подобно" при стойности между 0.5 и 1.

### **Функции за Принадлежност на Маневреност**

* **low**: Ниска маневреност при стойности до 30.
* **medium**: Умерена маневреност при стойности между 20 и 80.
* **high**: Висока маневреност при стойности между 60 и 100.

#### **Правила**

### **Правило 1: Ако разстоянието е много близко, рискът е много висок.**

Това правило казва, че ако два обекта са много близо един до друг, то рискът от сблъсък е много висок, независимо от другите фактори.

### **Правило 2: Ако разстоянието е близко, рискът е висок.**

Това правило предполага, че при "близко" разстояние между обектите, рискът от сблъсък е висок, но не толкова критичен, колкото при "много близко" разстояние.

### **Правило 3: Ако разстоянието е умерено, рискът е среден.**

При "умерено" разстояние, рискът от сблъсък е среден, което означава, че има потенциал за опасност, но ситуацията не е толкова непосредствена.

### **Правило 4: Ако разстоянието е далеч, рискът е нисък.**

Това правило посочва, че при голямо разстояние между два обекта, вероятността за сблъсък е ниска.

### **Правило 5: Ако относителната скорост е бърза И разстоянието е близко И (орбиталното сходство е подобно ИЛИ маневреността е ниска), тогава рискът е много висок.**

Това правило обединява няколко условия - висока относителна скорост, много близко разстояние и или подобно орбитално сходство, или ниска маневреност, за да установи ситуация с много висок риск от сблъсък.

### **Правило 6: Ако разстоянието е близко И относителната скорост е бърза, рискът е висок.**

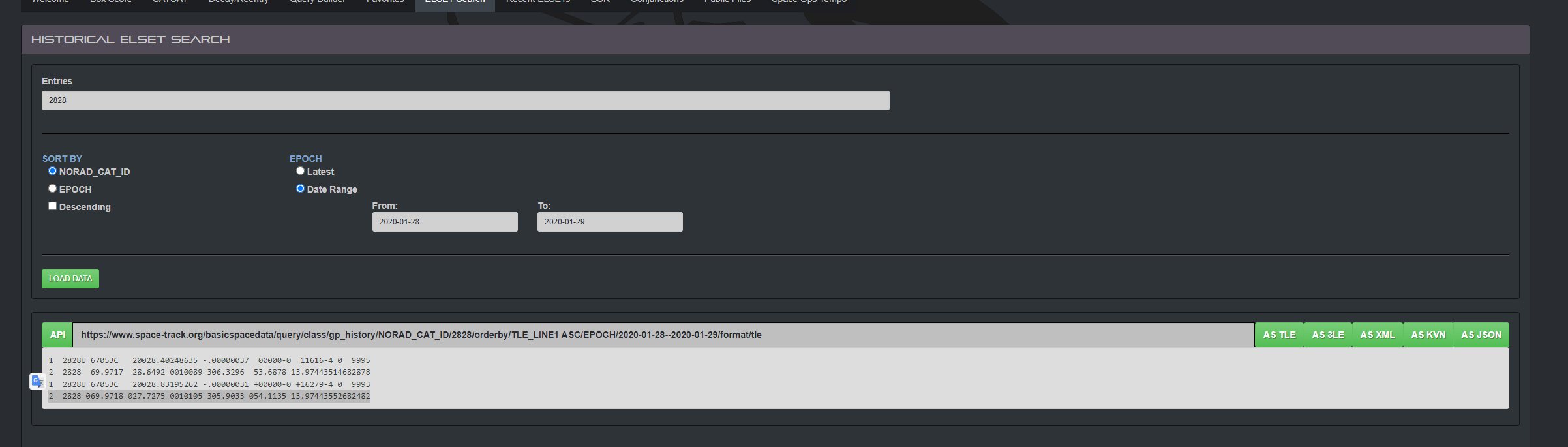
Това правило подчертава, че комбинацията от близко разстояние и висока относителна скорост увеличава рискът от сблъсък до висок.

### **Правило 7: Ако маневреността е висока И разстоянието е умерено И относителната скорост е умерена, рискът е нисък.**

Тук се предполага, че при висока способност за маневриране и при умерени стойности за разстояние и относителна скорост, рискът от сблъсък е нисък.

# Тестово изпълнение

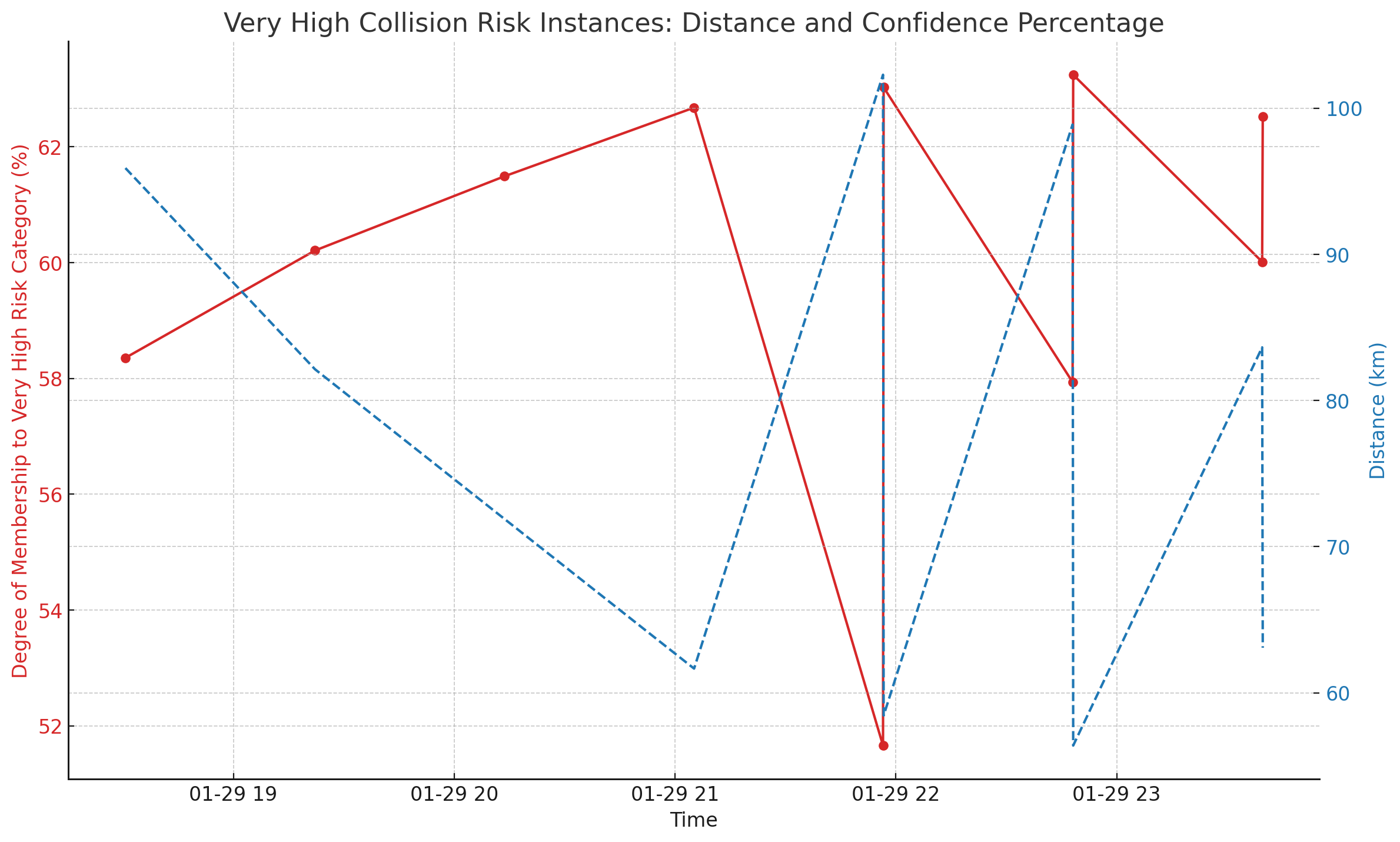
За да тестваме алгоритъма нека използваме статия от 2020.01.29, която ни съобщават за два сателита, които ще преминат критично близо един до друг на в 23:39ч UTC същия ден[2]. За целта ще ни трябва историческата TLE стойност с която може да се сдобием от space-track[3]:



*Фиг1. Интерфейс на space-track.org*

След като сме се сдобили с двете стойности ги заместваме в кода и поддаваме дата, за която искаме да проследим дали двата сателита биха имали колизия, в случаят това е 2020.01.29.

Нека сега разгледаме графиката, която показва случаите в който рискът от колизия е много висок(изобразено с червената линия) и дистанцията в определеният момент в км за съответният риск.



*Фиг2. Графика на “Висок шанс” за колизия на 2 сателита на 29.01.2020г*

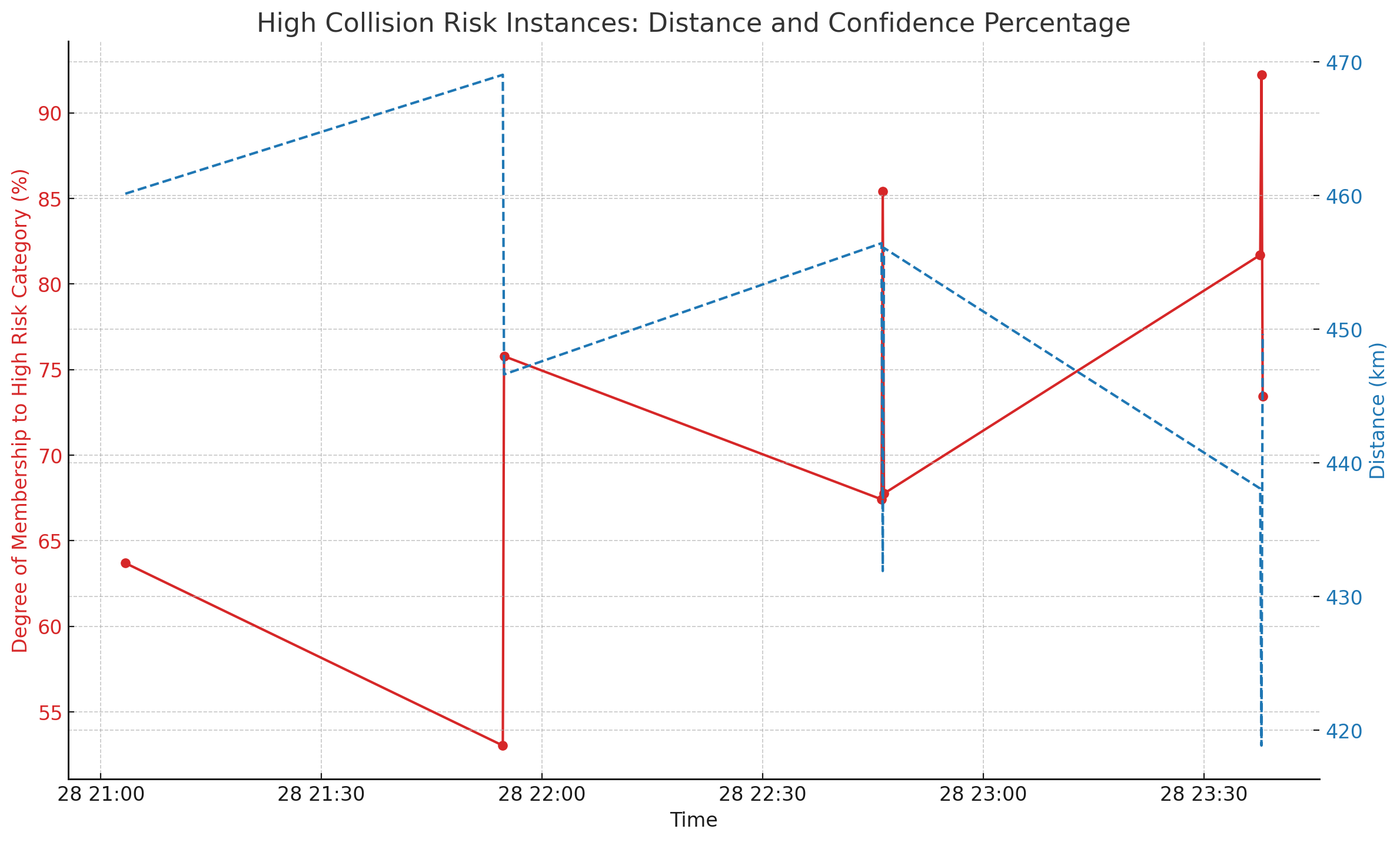
Виждаме, че има няколко случая за този ден, при които рискът е “Много висок”, един от тях е именно в споменатият час в статията 23:39 UTC, може да го сверим със стойностите в генерираният csv файл:



*Фиг3. Шанс за колизия на два сателита в 23:39ч UTC*

Виждаме че този “Висок шанс” за колизия е определен с увереност 70%, което звучи правилно, тъй като от статията разбираме, че астрономи са определили шанса за колизия 1:1000, което е доста висока стойност за сблъсък на два сателита.

Нека за всеки случай да проверим същите резултати за предишният ден(28.01.2020г). Както и предположихме след проверката нямаше нито един резултат за риск от колизия “Много Висок” има само 4 резултата с шанс за колизия “Висока” и те не са с колосални стойности дори:



# Заключение и възможно бъдещо развитие

Предложеният алгоритъм върши добра работа за начално наблюдение върху сателитите, обаче ако желаем да получим по-точни резултати за по-специфични сценарии ще трябва да въведем още правила, който трябва да бъдат надградени и съобразени след редица експерименти и анализ на резултатите за всеки от получените резултати.

Друг проблем, който срещаме е, че TLE данните за всеки сателит се обновяват най-често един път на ден, за някой сателити това става и по-рядко. Ако искаме да изградим надеждна система за следена на колизии на сателити в реално време ще трябва да преминем към по-иновативно решени, като например да следим сателитите с помощ на телескопи от земята. Предложеният алгоритъм е по-подходящ за телескопи от GEO сферата, където телескопите се движат с по-ниска скорост и генерално са по-статични и критични промени в орбитата им не са често явление.

# Използвани технологии и библютеки

* **Python**: Програмният език на който е написан кодът.
  + **csv**: За работа с CSV файлове.
  + **tkinter**: За създаване на графични потребителски интерфейси.
  + **customtkinter (ctk)**: Персонализирано разширение на tkinter.
  + **datetime**: За обработка на дати и часове.
  + **numpy (np)**: За числови изчисления.
  + **scikit-fuzzy (skfuzzy)**: За работа с размита логика.
  + **skyfield**: За астрономски изчисления.
  + **satellite**:Модулобработка на данни за спътници.

# Използвани литературни източници и Уеб сайтове

* [1] CelesTrak е уебсайт, предоставяш актуални и архивирани Two-Line Element Sets (TLEs) за проследяване на орбити на сателити: <https://celestrak.org/>
* [2] Spacenew събщава за потенциален сблъск между **Infrared Astronomical Satellite (IRAS)** и **Gravity Gradient Stabilization Experiment (GGSE) 4** в LEO сферата**:**  <https://spacenews.com/potential-satellite-collision-shows-need-for-active-debris-removal/>
* [3] База с данни пазеща исторически данни за сателити: <https://www.space-track.org/#/gp>