ISRO Ideation

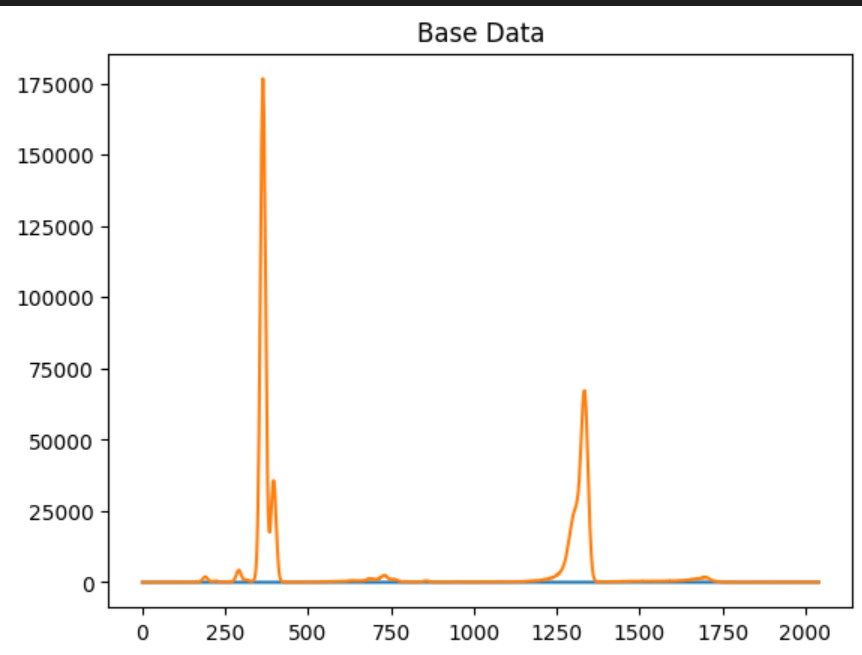
Below are some ideas which can be implemented in the ISRO problem statement. I have presented some sample programs as well

All programs can be found at the GitHub link <https://github.com/SuvrayanBandyopadhyay/ISRO-Ideation>. This link also contains some relevant research papers and other resources I have found.

# Finding number of XRF lines using python

For finding the number of XRF lines, we can use the SciPy library of python. SciPy has an in-built function called find\_peaks() which will return the number of peaks based on intensity and distance.

I have presented a sample code to demonstrate. Sample data has been collected from <https://data.mendeley.com/datasets/nkpmdtdkfw/1>



## Step 1: Importing necessary libraries

A black screen with green text

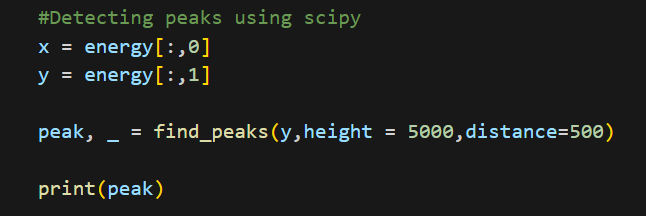
Description automatically generated

## Step 2: Processing data

A screen shot of a computer

Description automatically generated

## Step 3: Using SciPy to analyze data



## Step 4: Final detection

A screen shot of a computer code

Description automatically generated

A graph of a graph showing a blue line

Description automatically generated

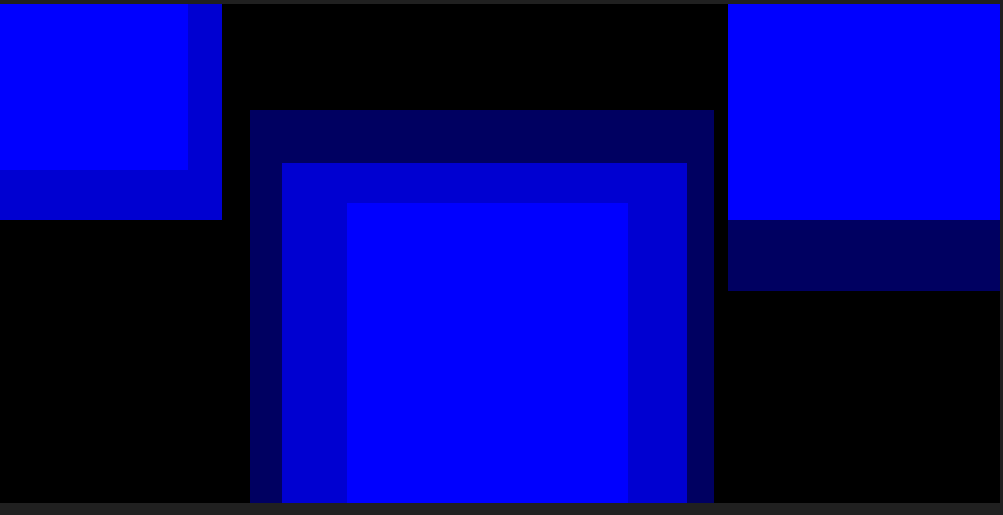
## Likelihood analysis and sub pixel resolution

Since the resolution of our data is bound by the resolution of the instrument which we are using we will need to implement sub-pixel resolution which will give us greater insights into the actual distribution of elements on the surface.

One way this can be achieved is described as follows:

We use 2 random made images for the sake of showing the method

Element 1 Map



Element 2 Map

A red rectangle and black rectangles

Description automatically generated

### We check which element has the highest probability at each element and accordingly set values

A computer screen shot of text

Description automatically generated

### Final visualization

A graph with different colored squares

Description automatically generated

The method is such that darker values correspond to a greater likelihood of finding the element at that point.

This method can be further improved on by analysing nearby pixels to get a much better image as described here (<https://www.researchgate.net/figure/llustration-of-sub-pixel-mapping-method_fig1_272183788>). In this method we decompose pixels based on the percentage abundances (If we have 16 pixels and 40% abundance of element A, then 6 pixels will be defined as class A. The position of these pixels depends on the fractional abundance of nearby pixels.

