

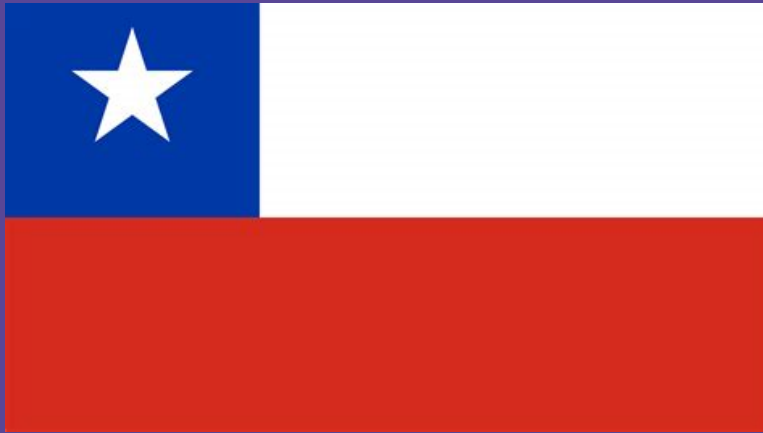
The background is a dark blue space-themed graphic. It features several concentric, thin white circular orbits. Scattered throughout are white stars of various sizes and shapes, some appearing as simple dots and others as multi-pointed stars. Four solid teal-colored circles of different sizes are positioned at various points along the orbits, resembling planets or moons. In the center of the image is a teal-colored map of the Earth, showing the continents of North and South America.

Classification on Sentinel 2 dataset - Copenhagen, Denmark

Google Earth Engine

Sugandh Bansal & Katherine Hansen

Introduction



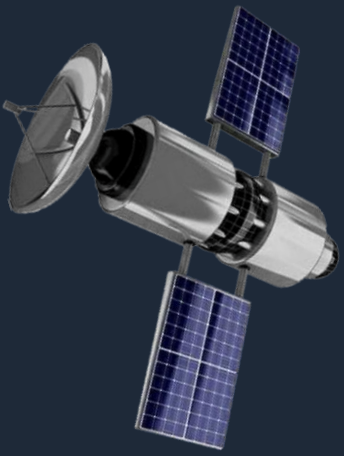
Goal

- To create something from scratch
- To learn collecting data
- To get challenged
- To implement our learnings, that we gain through these weeks in Saturday AI



Problem statement

- Classify satellite image with correct bands and getting right result

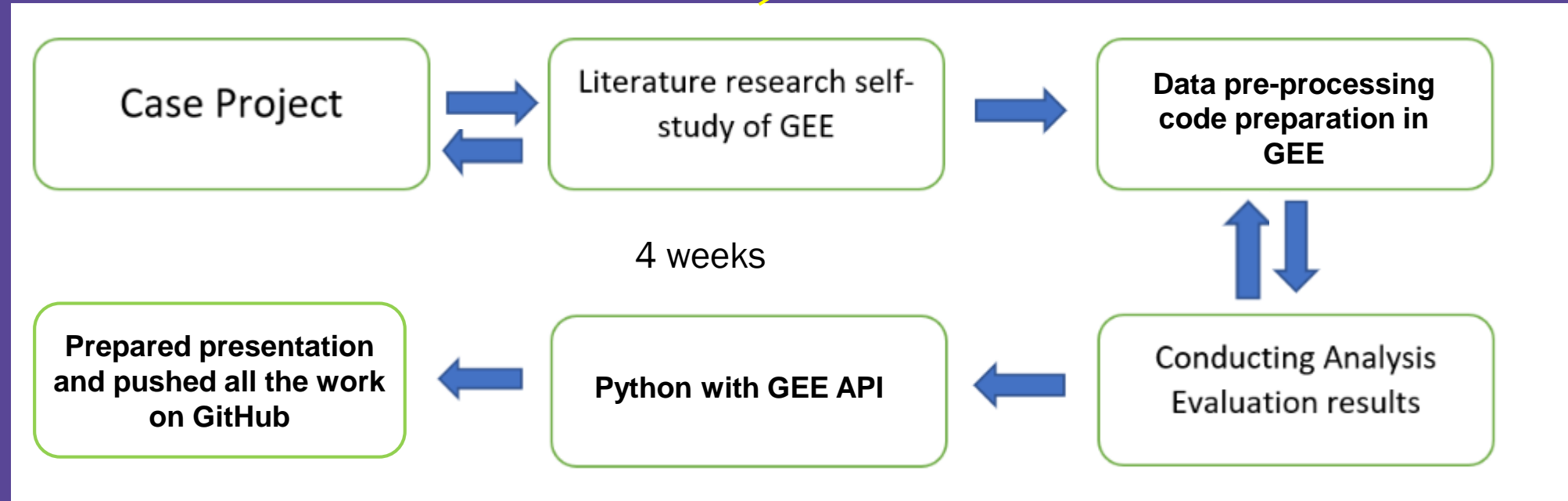


Data

We collected images from Sentinel 2 dataset available in Google Earth Engine, where we filtered data on the base of bands within the particular time frame, where we can get as much as possible cloud free images to get clear view.



Methodology



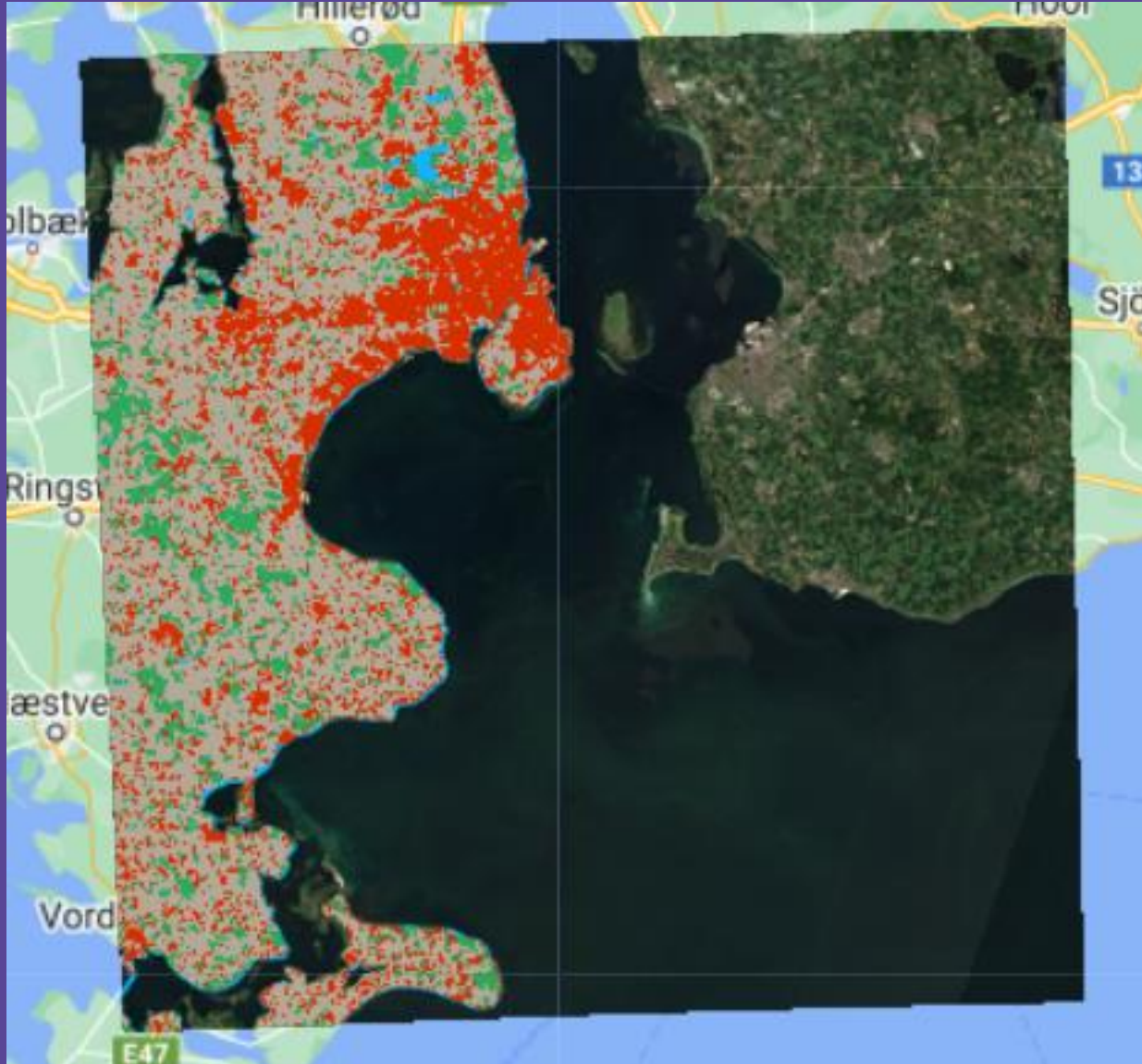
The general workflow for classification includes gathering training data, creating a classifier, training the classifier, classifying the image, and then estimate error with an independent validation dataset with the help of confusion matrix.

Challenges

- Limited time
- To understand Google Earth Engine platform
- To change JavaScript into Python
- To connect data from GEE to Python
- To find the right band to get the images of different classes defined in classification
- To find the right accuracy, right bands with confusion matrix



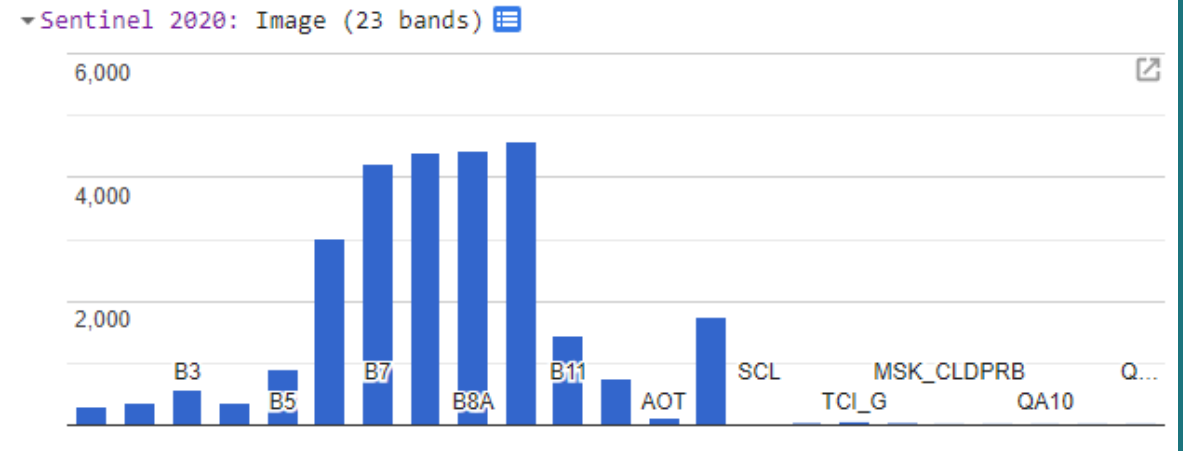
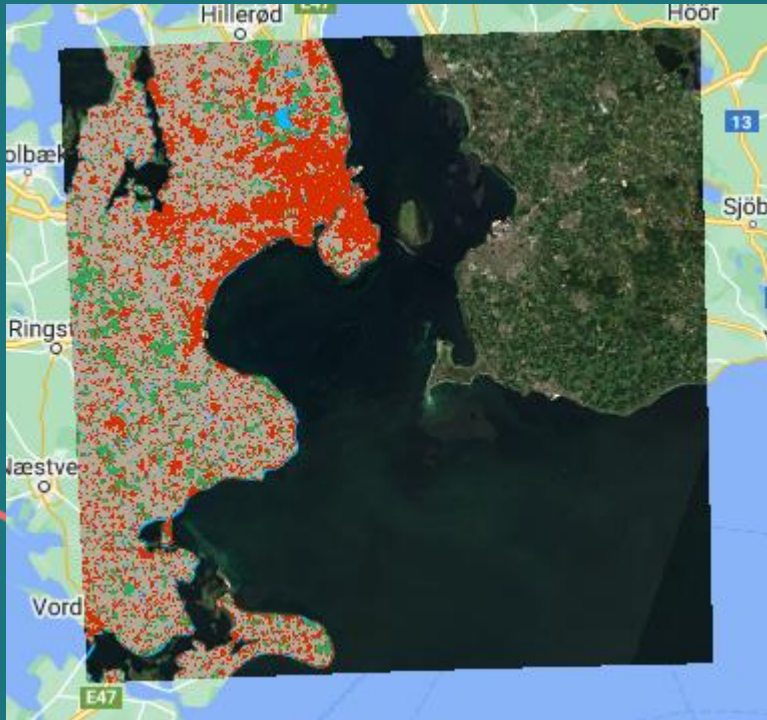
Classified Image



Blue – Water
Red – Urban
Green – Forest
Brown - Farmland

Band selection – trial 1

```
//Sample Imagery at Training Points to C  
var label = "Class";  
var bands = ["B2","B3", "B4","B8A"];  
var input = image.select(bands);//input
```



ConfusionMatrix:

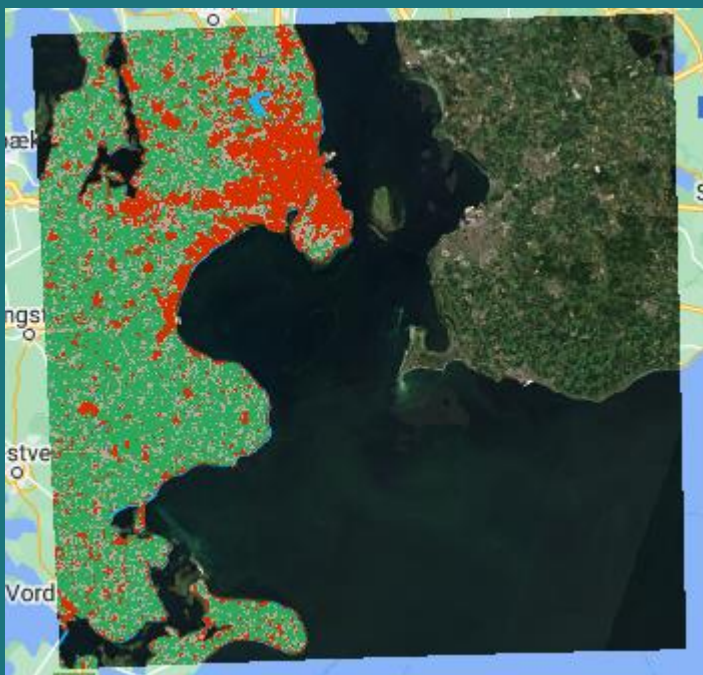
► `[[185,2,0,0],[0,293,0,1],[0,0,30,0],[0,0,0,51]]`

Overall Accuracy:

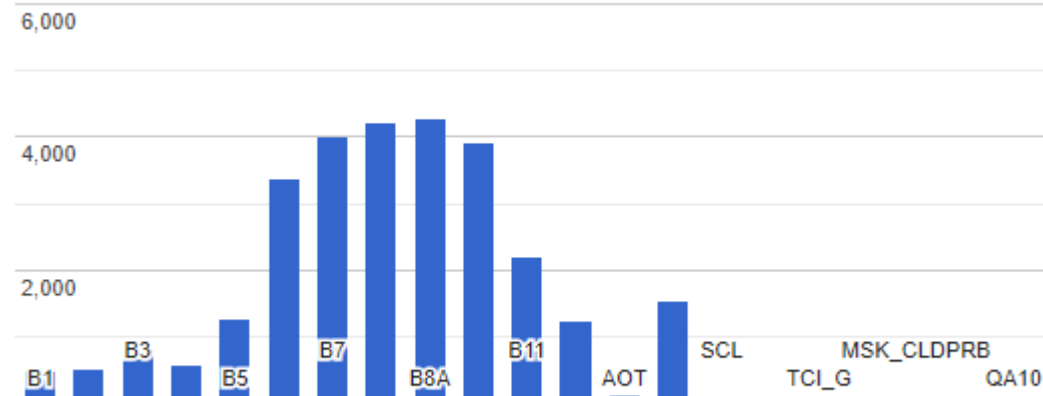
`0.994661921708185`

Band selection – trial 2

```
//Sample Imagery at Training Points t  
var label = "Class";  
var bands = ["B6","B7", "B8","B8A"];  
var input = image.select(bands);//inp
```



▼ Sentinel 2020: Image (23 bands)



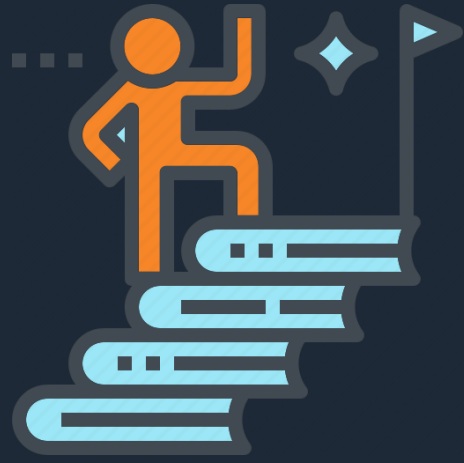
► Classification CART: Image (1 band)

ConfusionMatrix:

```
[[199,6,0,0],[1,297,1,0],[0,2,28,1],[0,1,6,45]]
```

Overall Accuary:

0.969335604770017



Achievements

- How to work with satellite images (eg. Landsat , Sentinel)
- How to use GEE for image classification.
- Better understanding of different image bands
- Confusion matrices can be used to assess the accuracy of supervised classifiers but should be used with caution
- How to calculate precision and recall
- Learned how to transform JavaScript into Python
- Working with different models for supervised classification



Tools that helped to build the project

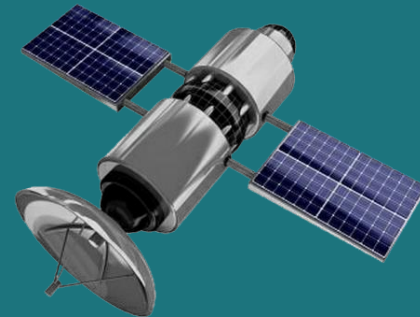
- Google earth engine : JavaScript



- Anaconda and Colab : Python



- Satellite data : Sentinel dataset with API tool



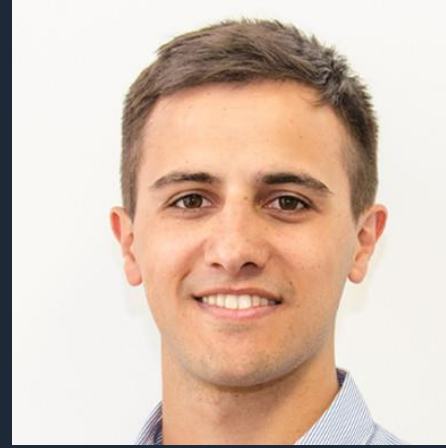
JavaScript to Python

- We used the same code in colab and in Jupyter notebook and changed all the codes into python
- Google earth engine developer code link..
<https://code.earthengine.google.com/?scriptPath=users%2Fsugban%2FCoral%3AClassification%20-%20Denmark>
- Colab link with Python script
<https://drive.google.com/file/d/1-4zX2XMPs3JBD1NVW8C6rduWjIZF9hHm/view?usp=sharing>

Next Step:

1. To find out the right accuracy
2. To classify dataset with time-lapse and observe the changes in the classified areas
3. Keep learning different tools and models in ML and AI





Thank you

