# Schneider Electric Hackathon

## Data Science Challenge - DT42

November 2022


**José Luz**

Msc in Energy for Smart Cities

jose-h-luz/



**Catarina Almeida**

Msc in Sustainable Energy

catarina-inls-almeida/



**Bendiks Herbold**

Msc in Energy for Smart Cities

bendiks-herbold/

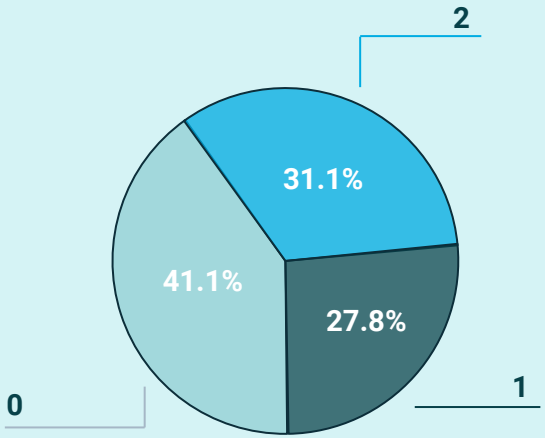


Fig. 1: Distribution of the classes in the training dataset

# Class Imbalance

01

## Impact on the model

The model predicted the classes with fewer observations ( 1 and 2) worse because it has fewer instances to train on.

02

## Over-sampling

To solve the imbalance it was use **over-sampling**: duplication of minority class observations.

03

## Results

The results of over-sampling can be seen in figure 2.

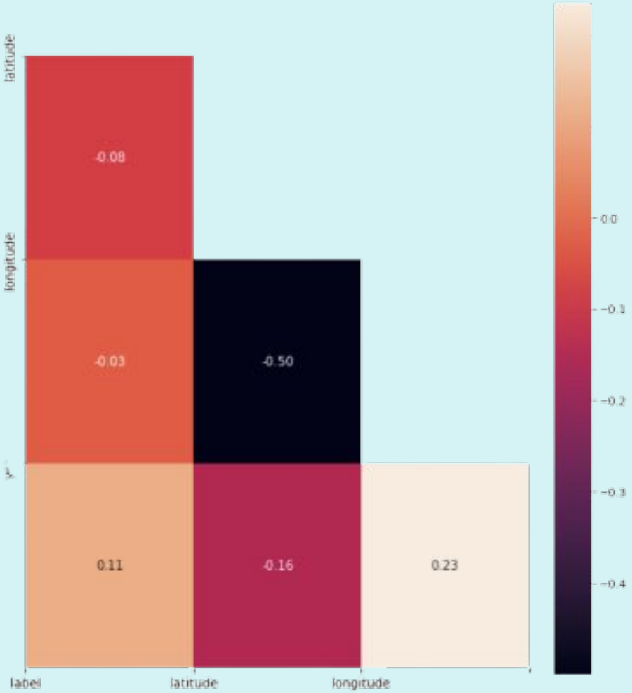


Fig. 2: Attributes correlation matrix

# Correlation Coefficients

01

## Latitude, longitude and year

As it can be seen in figure 2, the influence of these features on the label can be neglected.

02

## Deforestation pictures

Therefore, it was concluded that the only feature that should be consider is the deforestation pictures.

```
In [4]: dt_train.shape
Out[4]: (2106, 5)

In [5]: dt_train.head
Out[5]: <bound method NDFrame.head of       label  latitude   longitude  year
        example_path
        0         0 -2.051853  111.826093  2001  train_test_data/train/1297.png
        1         2 -1.989349  105.309496  2013  train_test_data/train/1199.png
        2         0  1.223256  100.702217  2014  train_test_data/train/1348.png
        3         0 -2.342948  103.890226  2008  train_test_data/train/2214.png
        4         0 -0.126555  101.758175  2011  train_test_data/train/2220.png
        ...     ...       ...         ...   ...                             ...
        1659      1 -6.500508  138.704721  2015  train_test_data/train/1627.png
        1663      1 -2.950291  133.193605  2015  train_test_data/train/2222.png
        1678      1 -2.251886  114.664116  2016  train_test_data/train/1750.png
        1705      1 -0.978231  110.183019  2015   train_test_data/train/352.png
        1711      1  0.443397  112.200163  2012  train_test_data/train/1486.png

        [2106 rows x 5 columns]>

In [6]: images = []

        for image_path in dt_train['example_path']:
            img = cv2.imread(image_path)
            images.append(img)

        images[0]=images[0]/255.00
        images = np.array(images)

        dt_train['images'] = images

In [7]: dt_train.shape
Out[7]: (2106, 6)

In [8]: # Check if the labels are unbalanced
        ### your code here
        balance_clases = dt_train['label'].value_counts()

        print(balance_clases)

        # Class balance graph
        balance_clases.plot.pie()

        0    860
        2    658
        1    588
        Name: label, dtype: int64
```

# Results

**01**

## Deep Learning algorithm

After testing several algorithms, the best results were obtained with **Convolutional Neural Network**.

**02**

## Hyperparameters

The hyperparameters, including the number of filters in the Neural Network, were adjusted to obtain better results.

**03**

## Analysis

To analyse the final results, it was calculated the **F1-score metric**. As it was expected , *class 0* had better results, since more data was provided for this class.