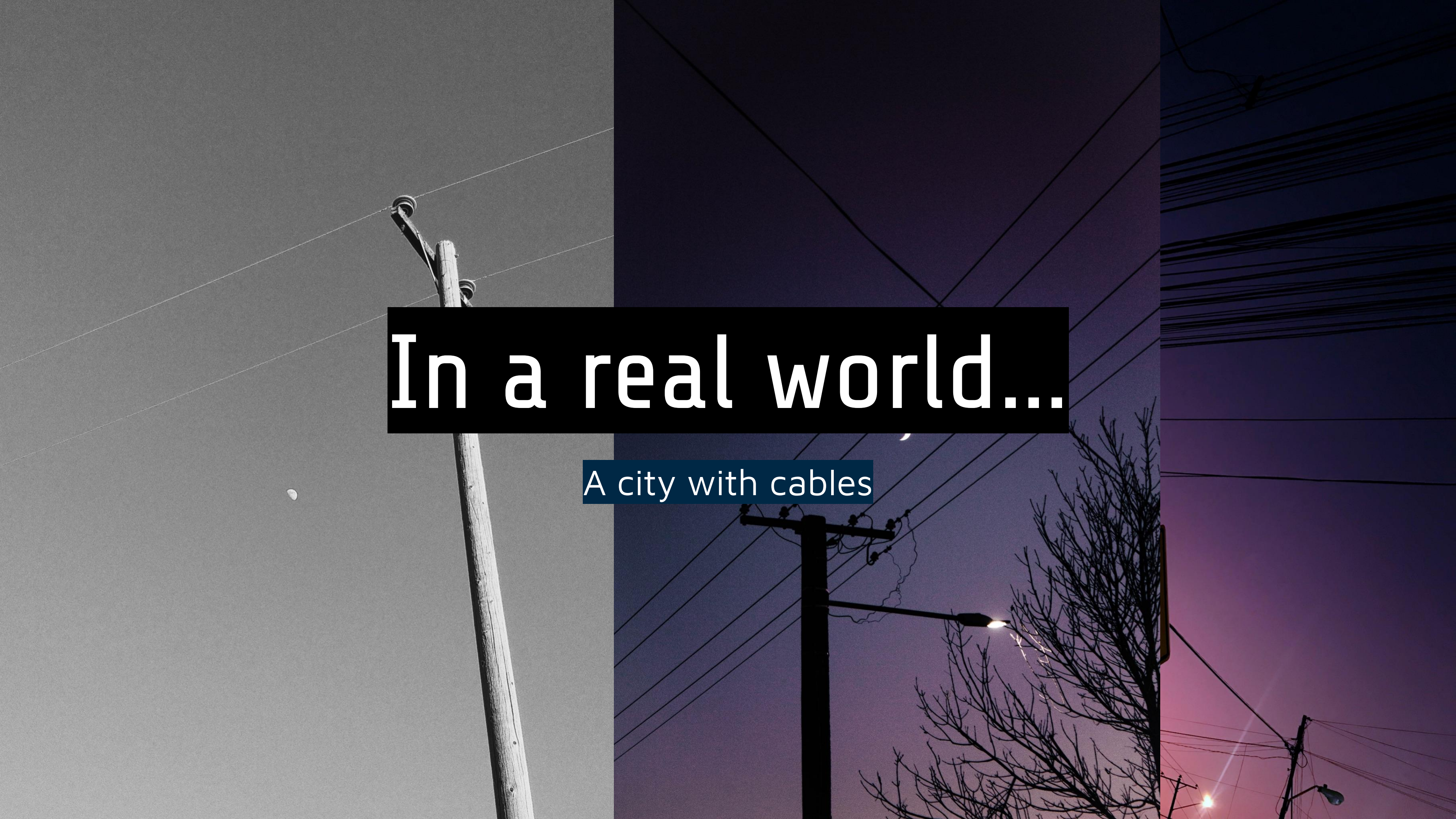
Segmentation of LIDAR data

LSSDS-2022

By Work-in-Progress Team

In an ideal world…

A city without cables


In a real world...
A city with cables

The problem

Identify trees near cables proximity between trees and power lines represents a significant fire threat in urban and rural areas, while managing millions of kilometers of power lines in the world is a challenging task.

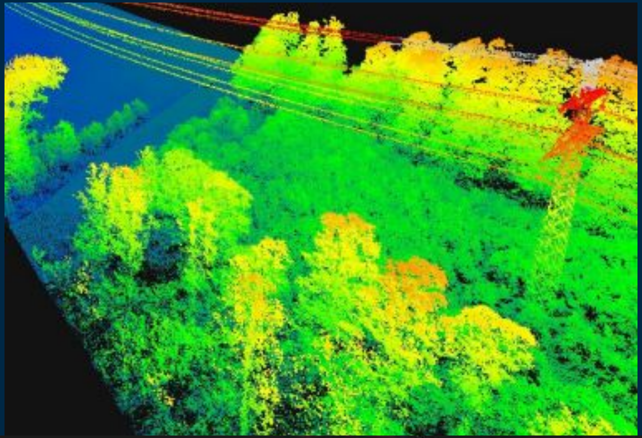
It would be great to have a tool to automatically detect trees in proximity to cables to perform optimal maintenance!

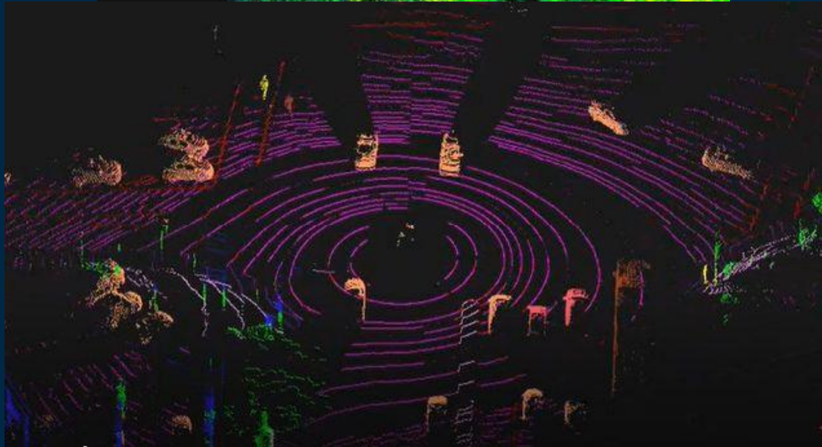
Francisco Förster

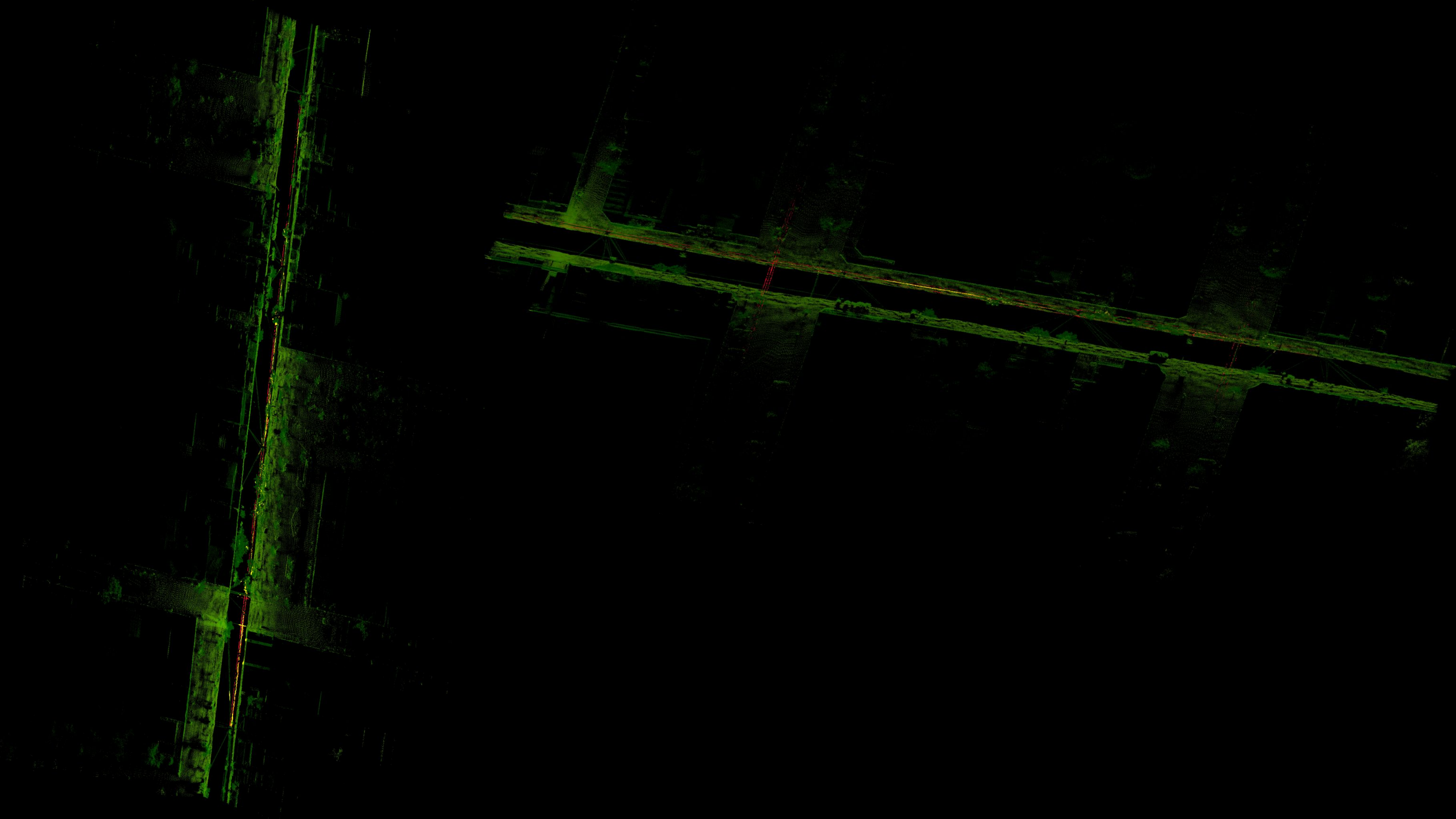
Light Detection And Ranging (LiDAR)

LiDAR is a method for determining ranges (variable distance) by targeting an object or a surface with a laser and measuring the time for the reflected light to return to the receiver.

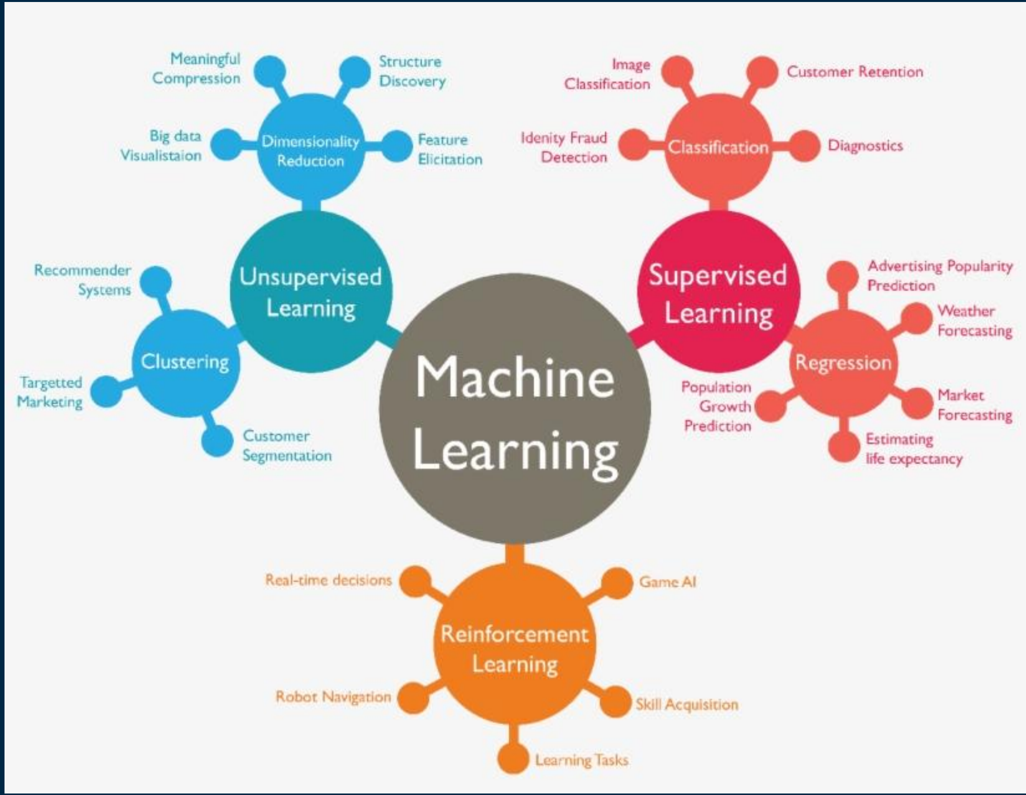
Data they produce contains 19 features







Machine Learning: Supervised

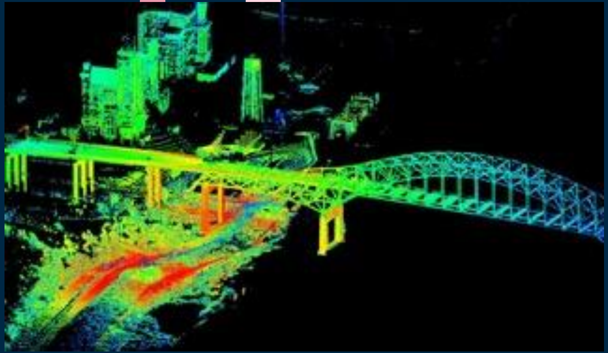


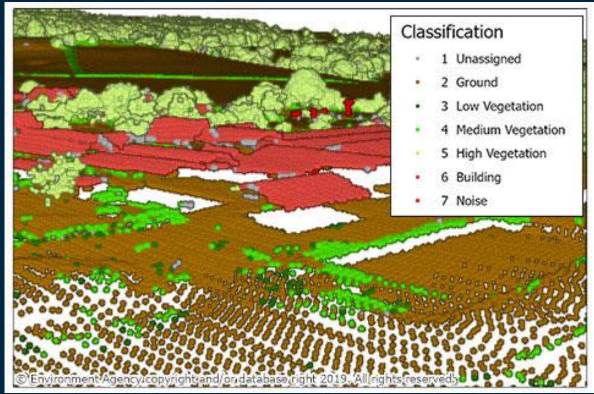
What Is Supervised Machine Learning (SML)?

- Also commonly referred to as supervised learning
- Uses labeled data sets to train algorithms
- Those algorithms are then used to accurately classify test data or predict outcomes
- The labeled data sets contain input variables (features) and output variables while the data we want to classify just has input data
- A segmentation classification problem is when we are assigning a label to each individual data point
- The classification labels are the outputs we want!

How LiDAR Relates to SML

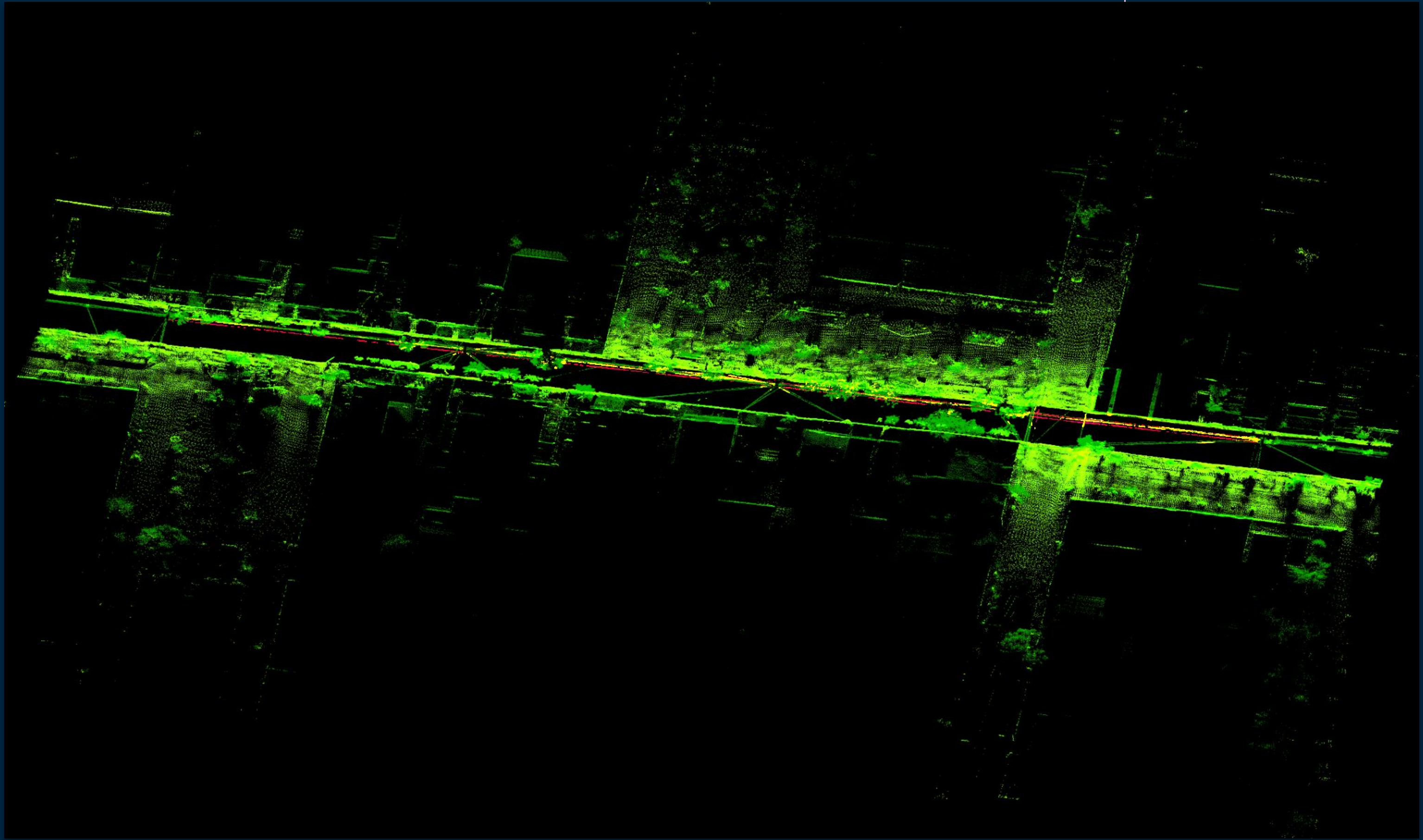
- This project is a segmentation problem since we are assigning a label classification to each individual point
- Project contains both processed and unprocessed data:
    - Processed data contains data points that are already classified
    - Unprocessed data does not have classified points already
- We can use the processed data to train a model (and compute features)
- Then we can use the trained model to classify the points of the unprocessed data
- Properly classifying the data points allows us to check for potential hazards, such as cables near trees!



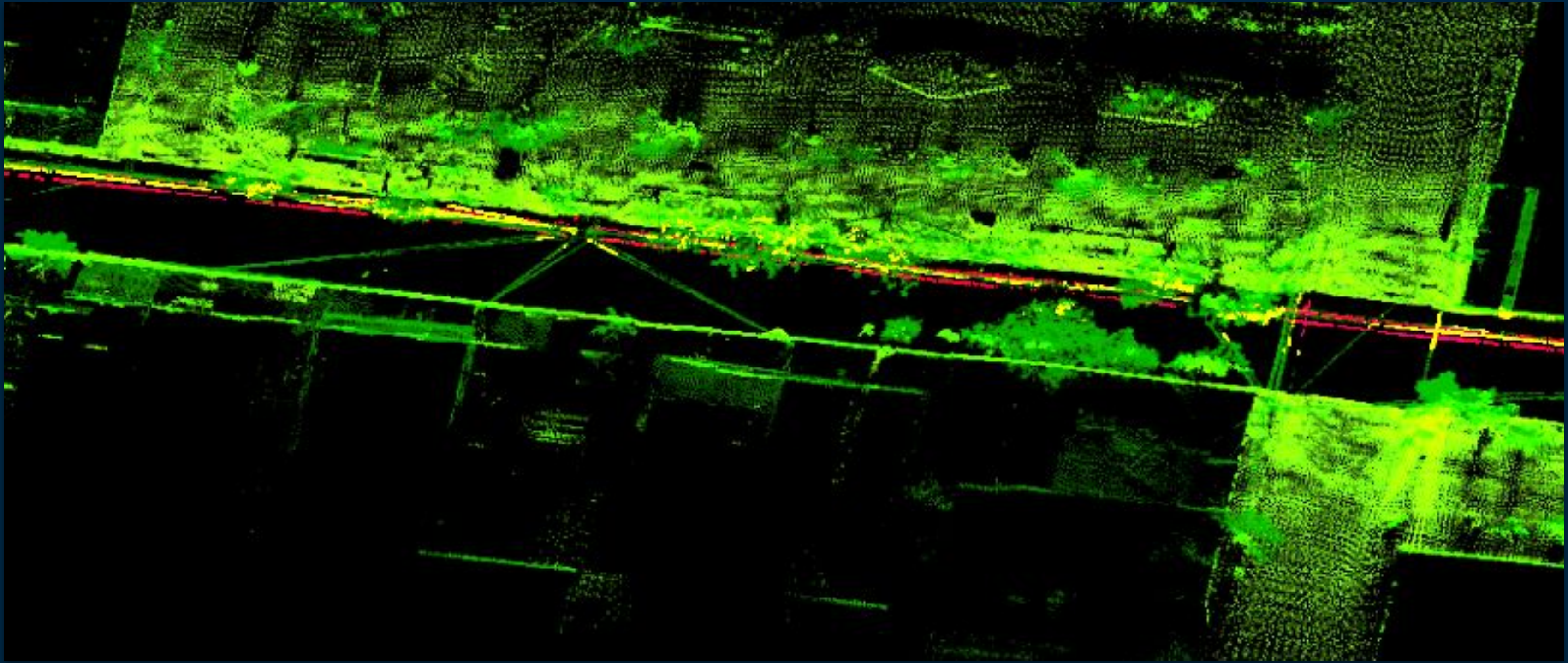


LiDAR Labels

- If we are classifying LiDAR data, we must know the labels!
- The labels are:
  - 2: Ground
  - 3: Low Vegetation
  - 4: Mid Vegetation
  - 5: High Vegetation
  - 11: High Power Cable
  - 14: Post
  - 26: Critical Zone
  - 27: Dangerous Zone
  - 29: Telecommunication Cable

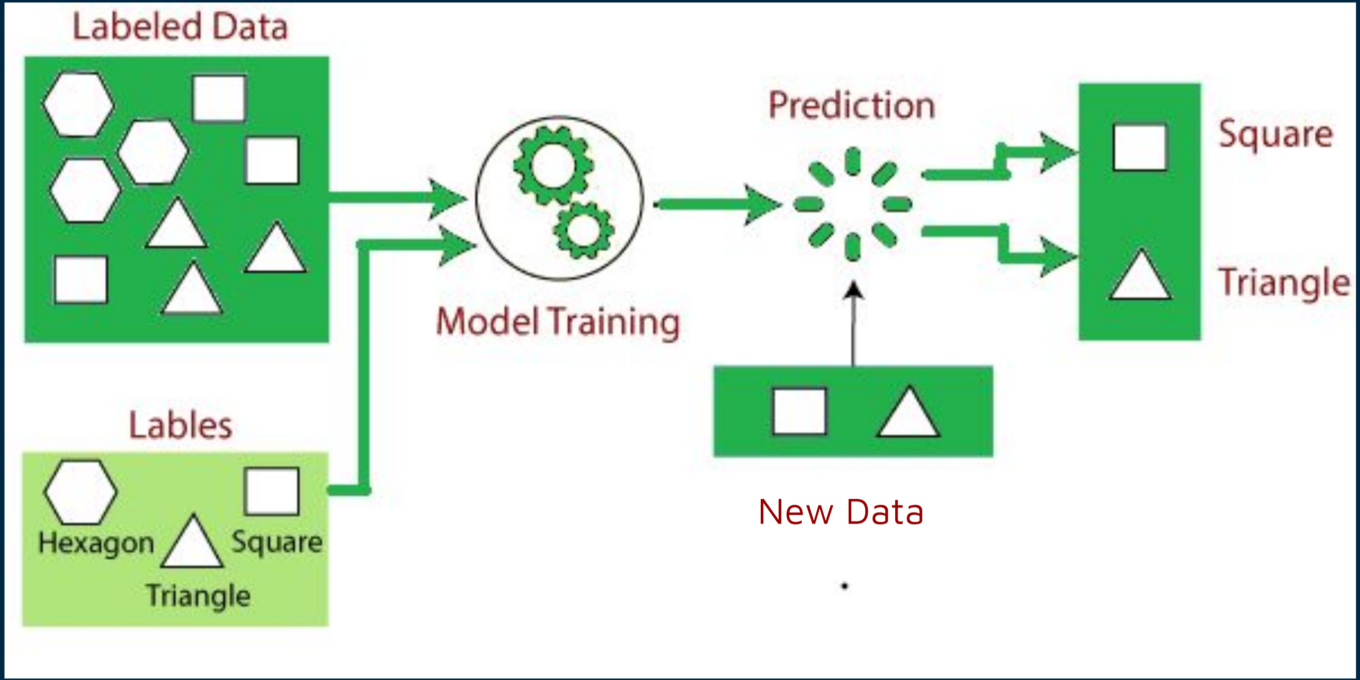


Here is one image of LiDAR data; it is a lot of data so let’s zoom in.



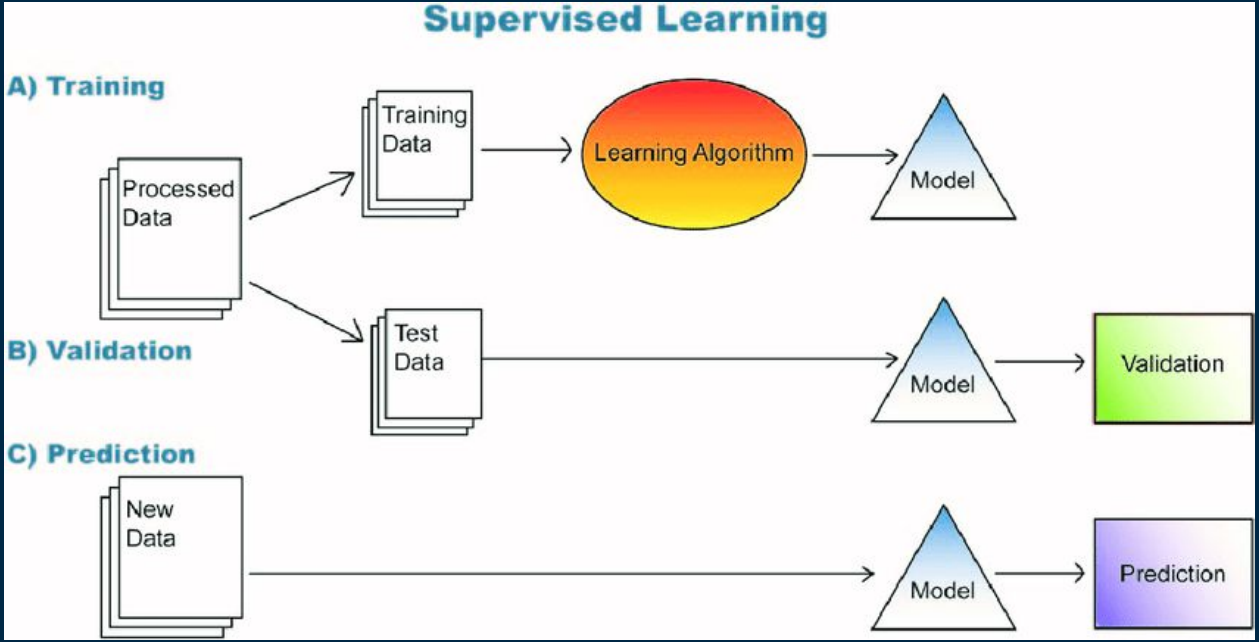
Green/Olive: Low, mid, high vegetation; Red: Communication cables; Yellow: Electrical cables

SML Overview



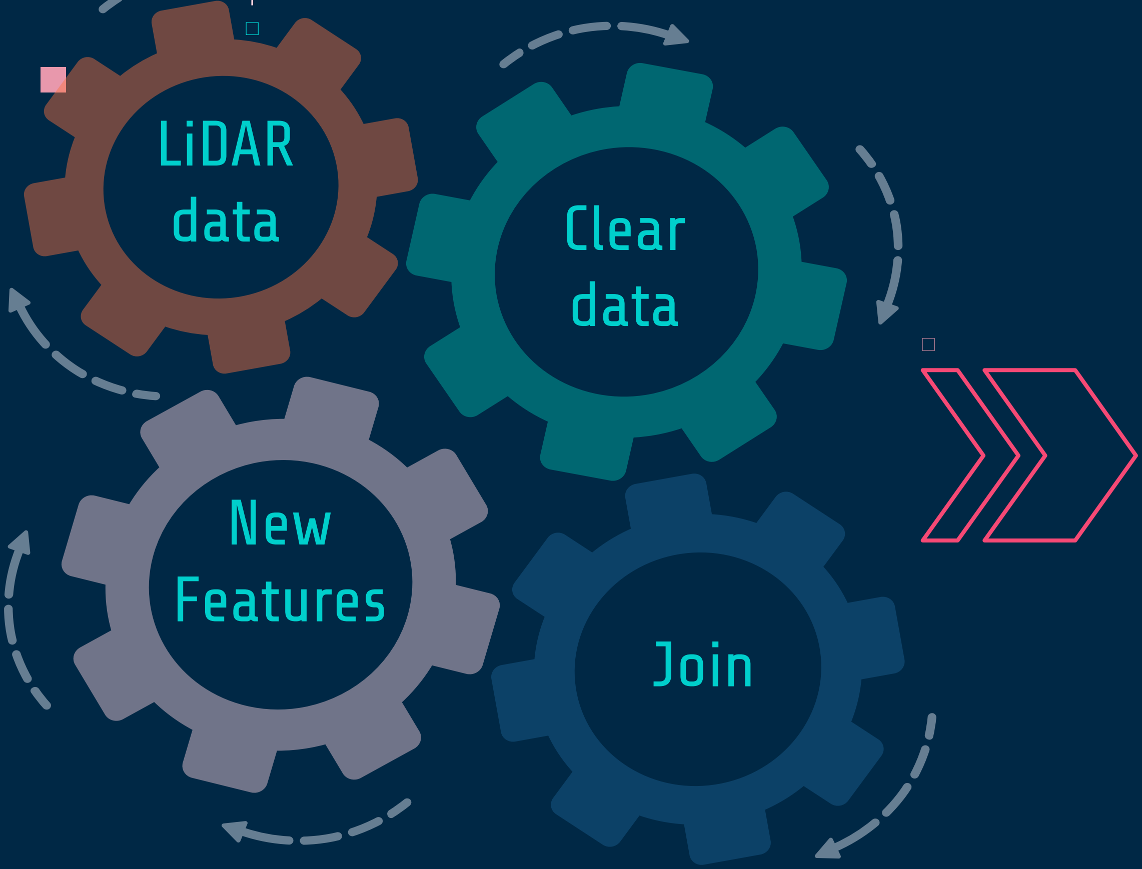
Model training and then using the model with new data

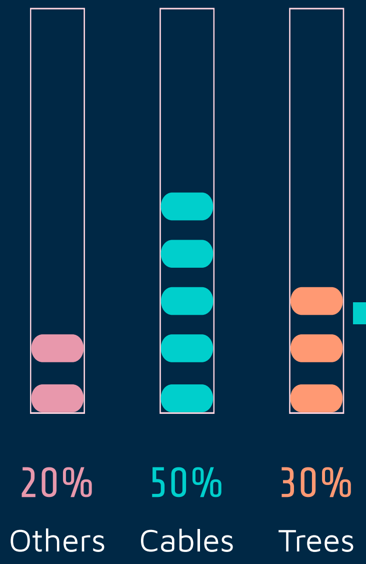
SML Overview



Using this idea, we developed our plan to deal with our specific LiDAR data!

We plan to do


LiDAR data
Clear data
New Features
Join


20%
Others
50%
Cables
30%
Trees

Assumption

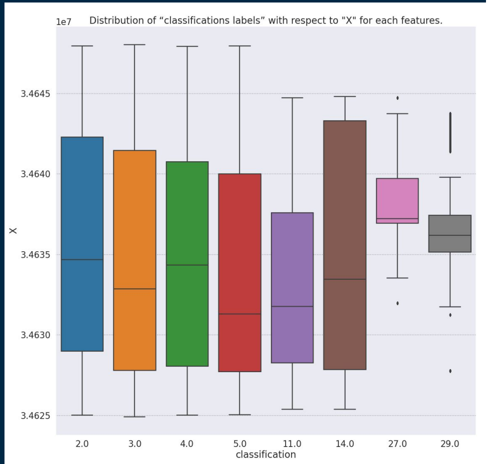
The height of trees (high-vegetation) and cables is similar

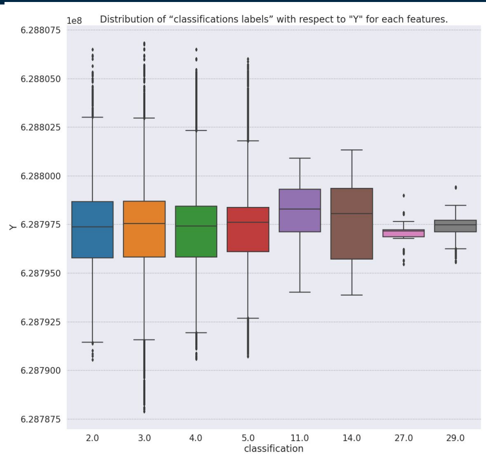
The distance between posts is similar or proportional.

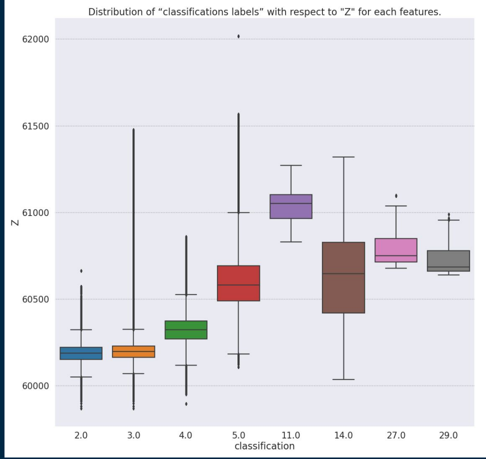
The intensity with which it is received can be useful to distinguish between features.

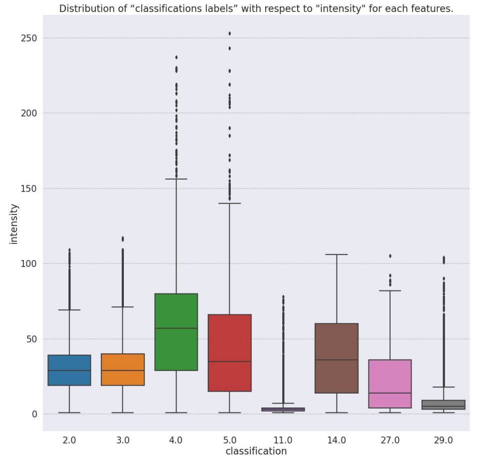
Descriptive statistics

```
classes = {
2: 'ground',
3: 'low vegetation',
4: 'mid vegetation',
5: 'high vegetation',
11: 'high power cable',
14: 'post',
27: 'dangerous zone',
29: 'telecommunication cable'}
```

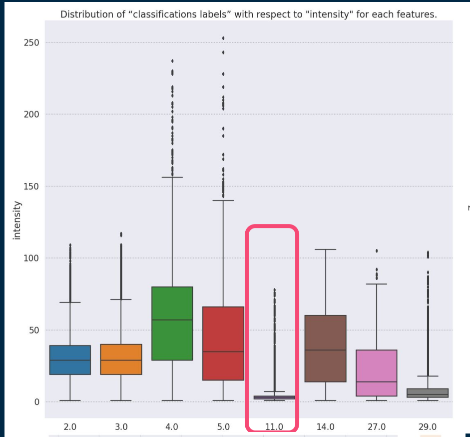


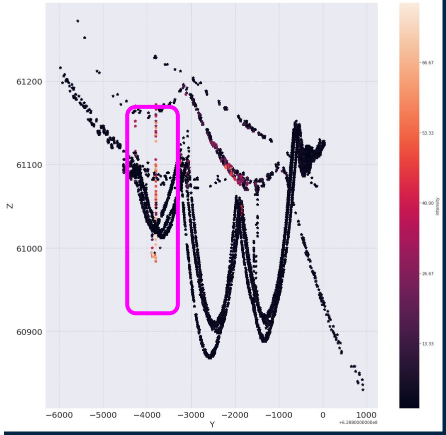


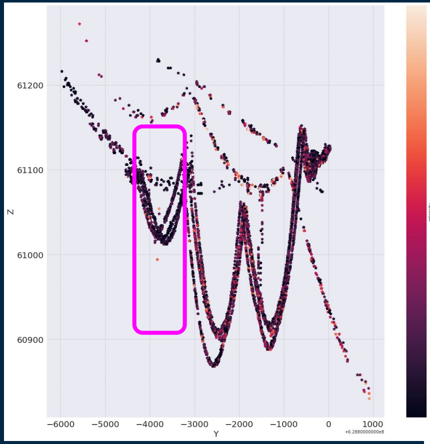


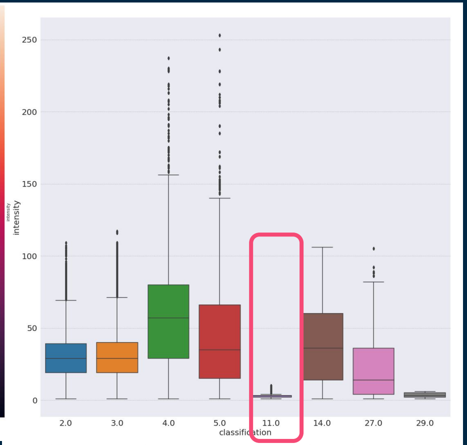


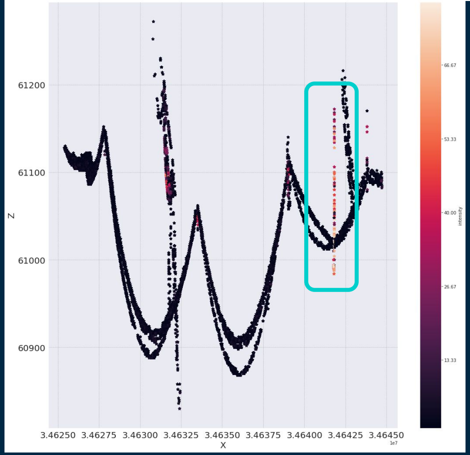
Data cleansing



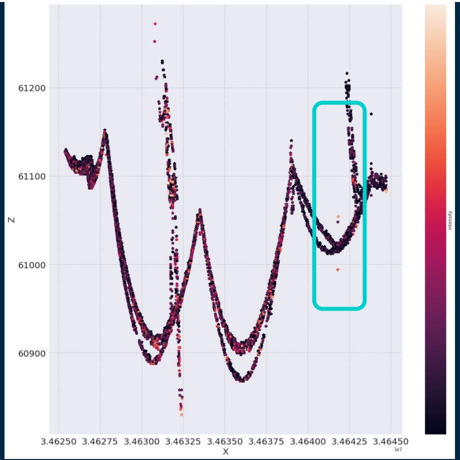




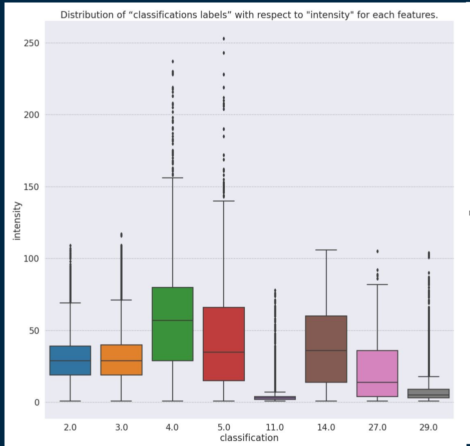


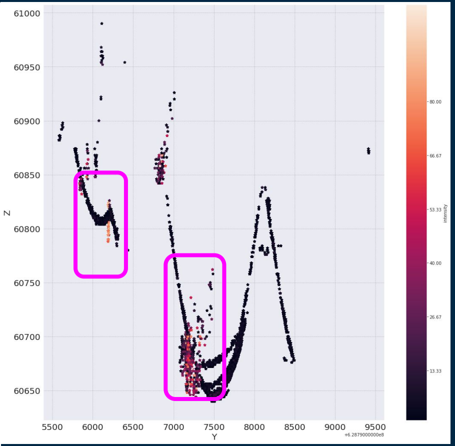


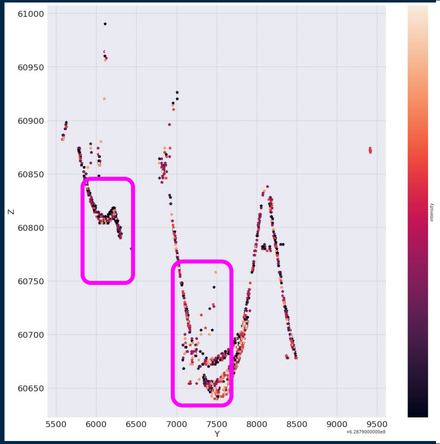
HIGH POWER CABLE

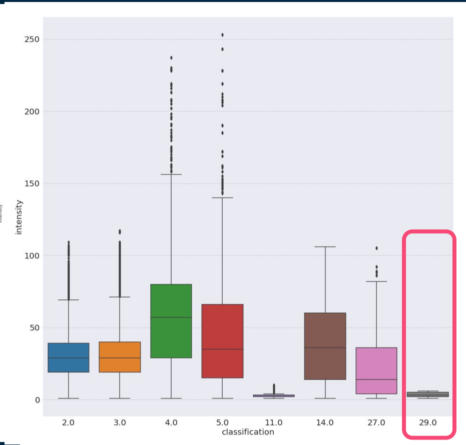


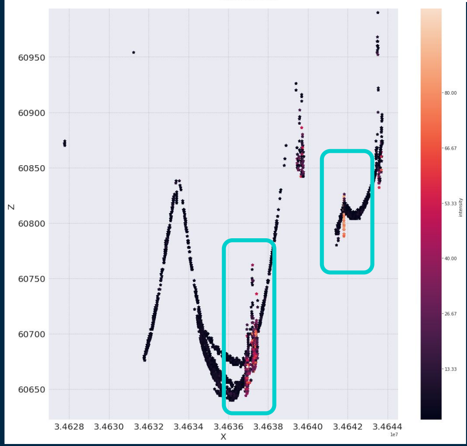
Data cleansing



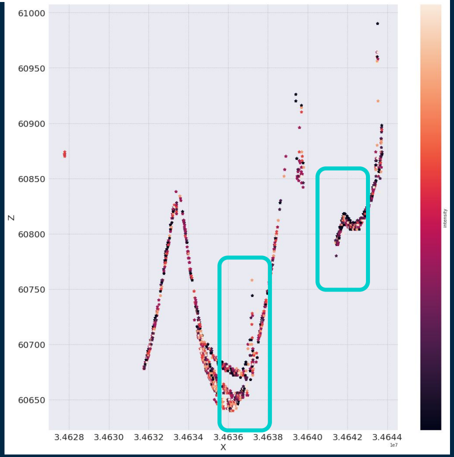




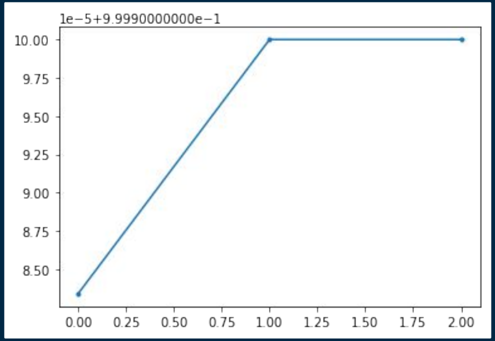




TELECOMMUNICATION CABLE

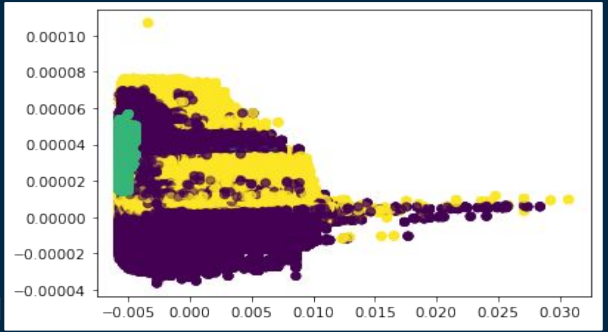


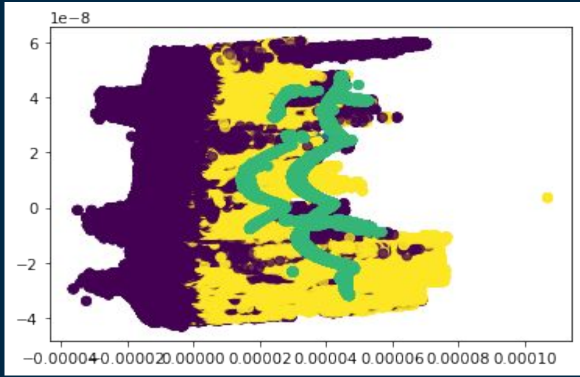
PCA

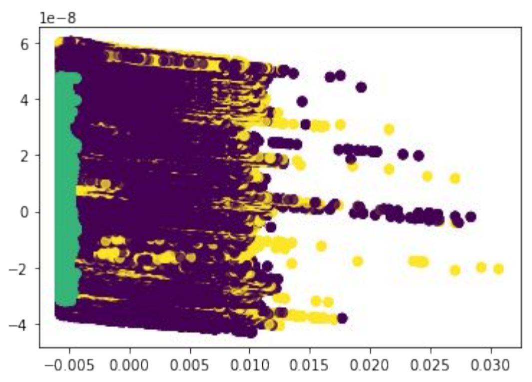


- Have 4 components ['X', 'Y', 'Z', 'intensity'].
- Tested with n_components=3

- Resulting that explained_variance_ratio is



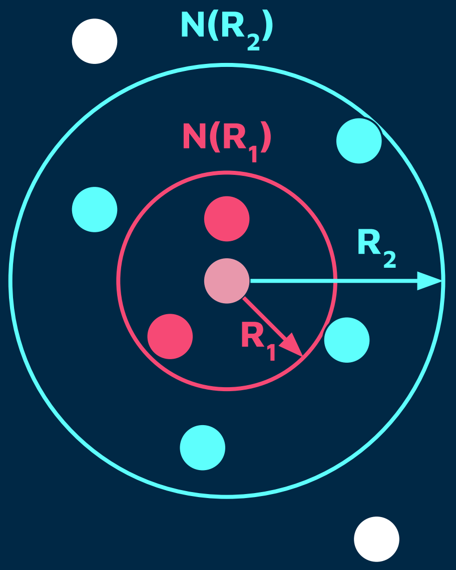




New Feature: the Geometry

The distribution of points around a given point can help the model identify the object

- If the point is part of a Cable → a line
- If the point is part of a Wall → a plane
- If the point is part of a Tree → a clump



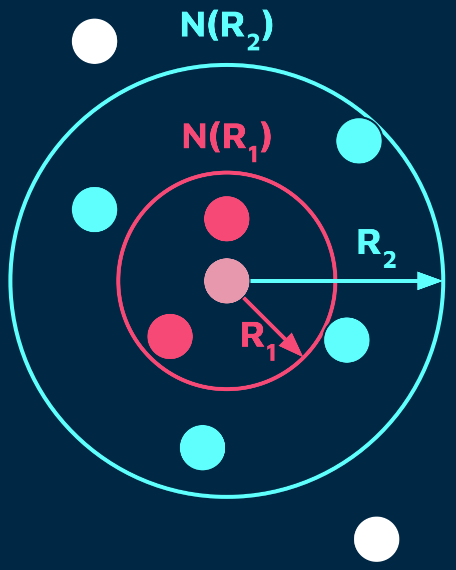
New Feature: the Geometry

Draw a sphere with a given radius R around this point and count the number of neighbor N(R) within this sphere.

Different geometries will have different N(R) as function of R

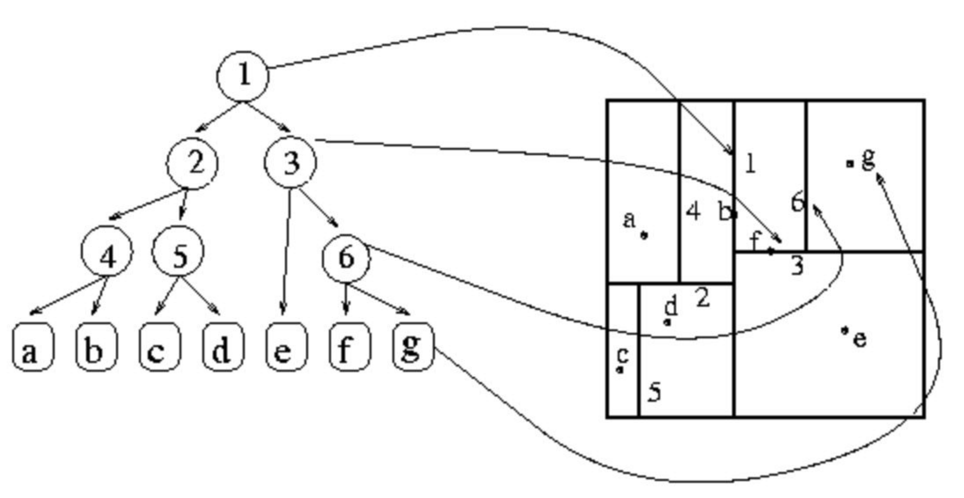
- If the point is part of a Cable → a line → $N(R) \propto R$
- If the point is part of a Wall → a plane → $N(R) \propto R^2$
- If the point is part of a Tree → a clump → $N(R) \propto R^3$

Approximate $N(R) \propto R^{\alpha}$, the slope α tells the geometry around a given point.



New Feature: the Geometry

Use K-D tree and the nearest neighbor (NN) algorithm to compute N(R)



The concept of the K-D tree is to partition the space into tree structure to improve the NN search

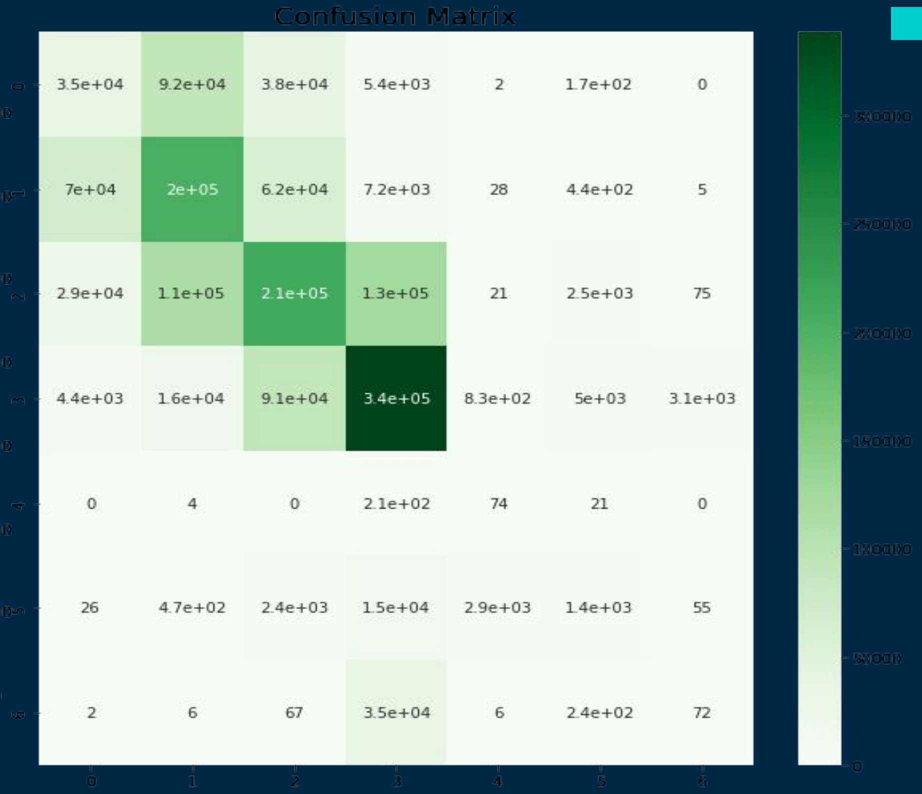
Supervised Classification

- We have data from five streets
- The idea is to use 4 streets to train the models.
- The fifth street will be used to test the models.

Some Results – kNN

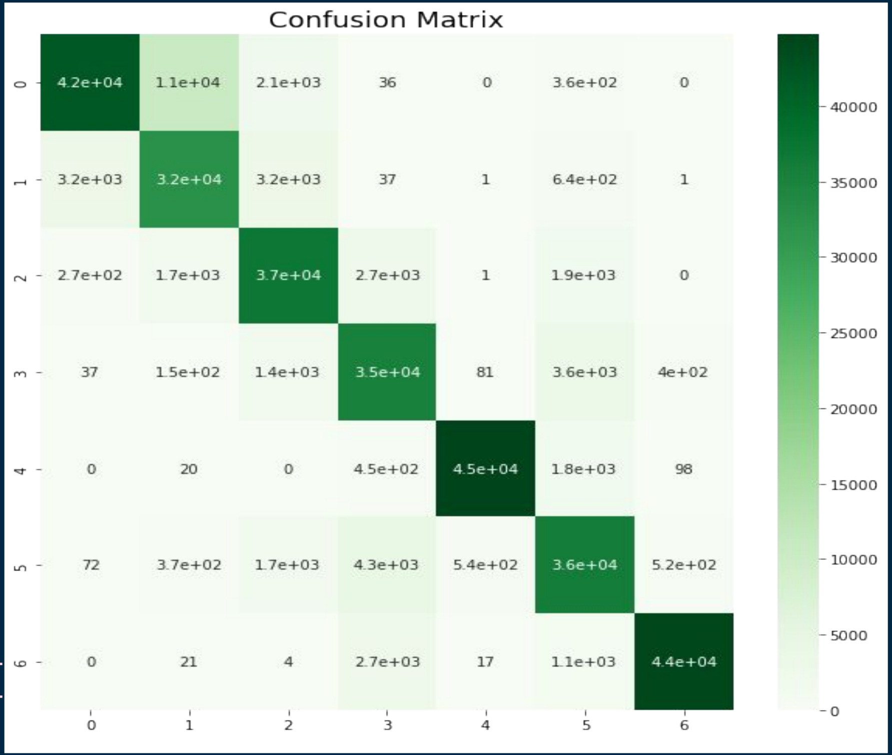
Train

Test

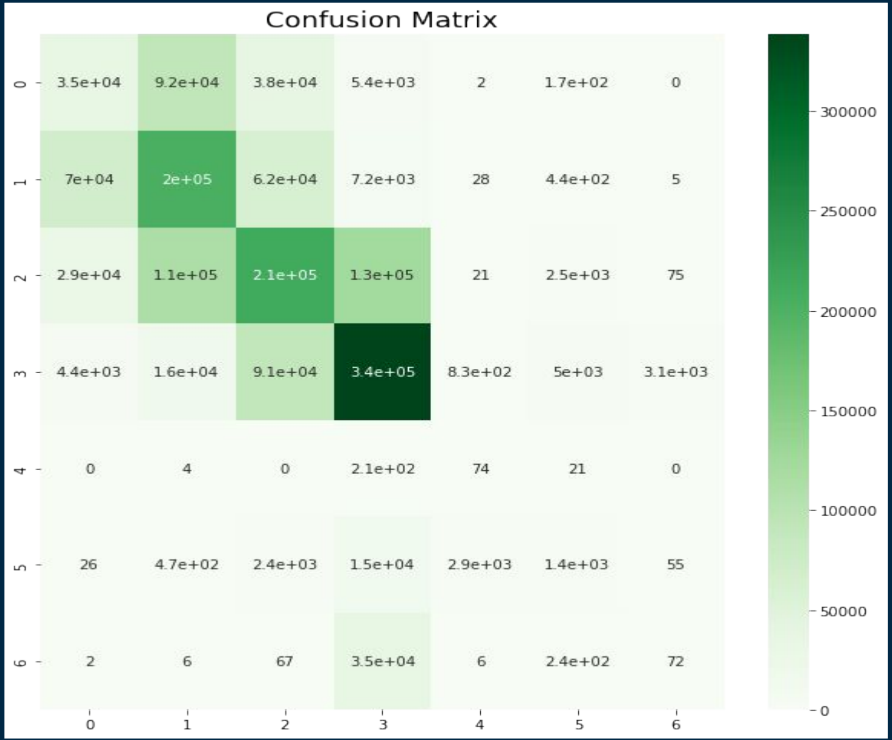


Random forest

Train



Test



- Results from deep neural network is not so different
- Classification report is poor for all classifiers.
- All models seem to overfit

What if we consider just one street?

Classification report:

	precision	recall	f1-score	support
2.0	0.87	0.34	0.48	48551
3.0	0.78	0.95	0.85	128427
4.0	0.95	0.96	0.96	231100
5.0	0.96	0.98	0.97	281792
11.0	0.99	0.95	0.97	1658
14.0	0.94	0.54	0.69	5323
29.0	0.98	0.70	0.82	8775
accuracy			0.92	705626
macro avg	0.92	0.77	0.82	705626
weighted avg	0.92	0.92	0.91	705626

What’s wrong?

- Undersampling made us lose significant amount of inform
- We did not have computing power
- The methods used in computing new features might not be rigorous enough

Conclusion

- For beginners it is difficult to define the path to follow to solve problems of this complexity.
- Needs more time, difficult to handle lidar data.