Software Contribution to the L-CAPE project

FERMILAB-POSTER-23-171-AD

Jasmine Tang - Santiago Canyon College

Advisor: Jason St. John

The technology, software, and data associated with L-CAPE were determined to be Fund Research Exclusion eligible, which removes export control related restrictions, as of August 3, 20202, by Paul Lauper Ellison, Export/Import Control Compliance Manager, Fermilab Office of General Counsel.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Community College Internships Program (CCI)

Introduction

The L-CAPE project utilizes asynchronous data from thousands of Linear Accelerator devices and applies data science techniques to detect anomaly of accelerator failure before the incident, as well as automatic labels of accelerator outages. The author describes her software and analysis contribution for the L-CAPE project this summer, as well as suggestions for future interns working on the project.

L-CAPE Data Guide

- Describes the L-CAPE devices (which are readings vs. settings, membership in RF stations, descriptive text, etc. The DataGuide live in a data_guidance.py class.
- DataGuide import function uses **pd.read_csv()** with custom parameters.
- Comes with demanded queries in the DataFrame.
- Relative path import in monolithic repository.

Raw Data Processing script

In order to provide machine learning with data that is comprised of the difference between reading devices and setting devices to make the model more resilient to changes made by the operators in the control group.

In *new_settings* branch of L-CAPE repository.

•Pseudo-code

```
• Glob all raw data files ending in .h5
• For each .h5 file in the glob, get the keys (which are the names of control system devices) from the .h5 files.
    • For each key in the keys collection
        • Check if that key is a setting device from the DataGuide,
        • if it is, add it to the collection if not already added, along with the file name that the key came from.
```

Uses multiprocessing pool of 8 cores, each `.h5` file is processed in a worker process.

The output is a collection represented as a dictionary, whose key A is the string representing the setting devices' name, and the value of that key is a set B of strings C. (So each value C in the set B signifies that key A was found in the file name of string C.)

Dimensional Reduction Model

For visualization and resolving of curse of dimensionality [https://umap-learn.readthedocs.io/en/latest/index.html]: better speed, preservation of data's global structure than t-SNE.

-> Interpretable results regarding the local and global structure.

For example: Distance between points in a cluster is preserved. Distance between clusters is also preserved.

-> Pickleable to later transform new/test data to learned space.

Clustering Model

UMAP does not preserve density, creates false tears, need to be attempt with care for clustering so we choose to rely on DBSCAN: Density-based clustering Spatial Clustering of Applications with Noise.

-> Sensitive to parameterized eps variable

Data Analysis: Minimal Population Cut Analysis

Since minimal count of labeled data, many classes had only 1 member -> cut analysis.

Figure 1 Plotting of the number of labels of each class.

Figure 2 Dataset reduction percentage: measures the percentage that is being lost after applying minimum count threshold to the dataset (shortage of labels)

Figure 3 Change in the quality of clustering of UMAP(side effect) reflected via the minimum count with constant `min_dist` and `n_components`

Misc

- Workflow
- Reading/Setting Device mismatch
- Timestamp Communication mismatch

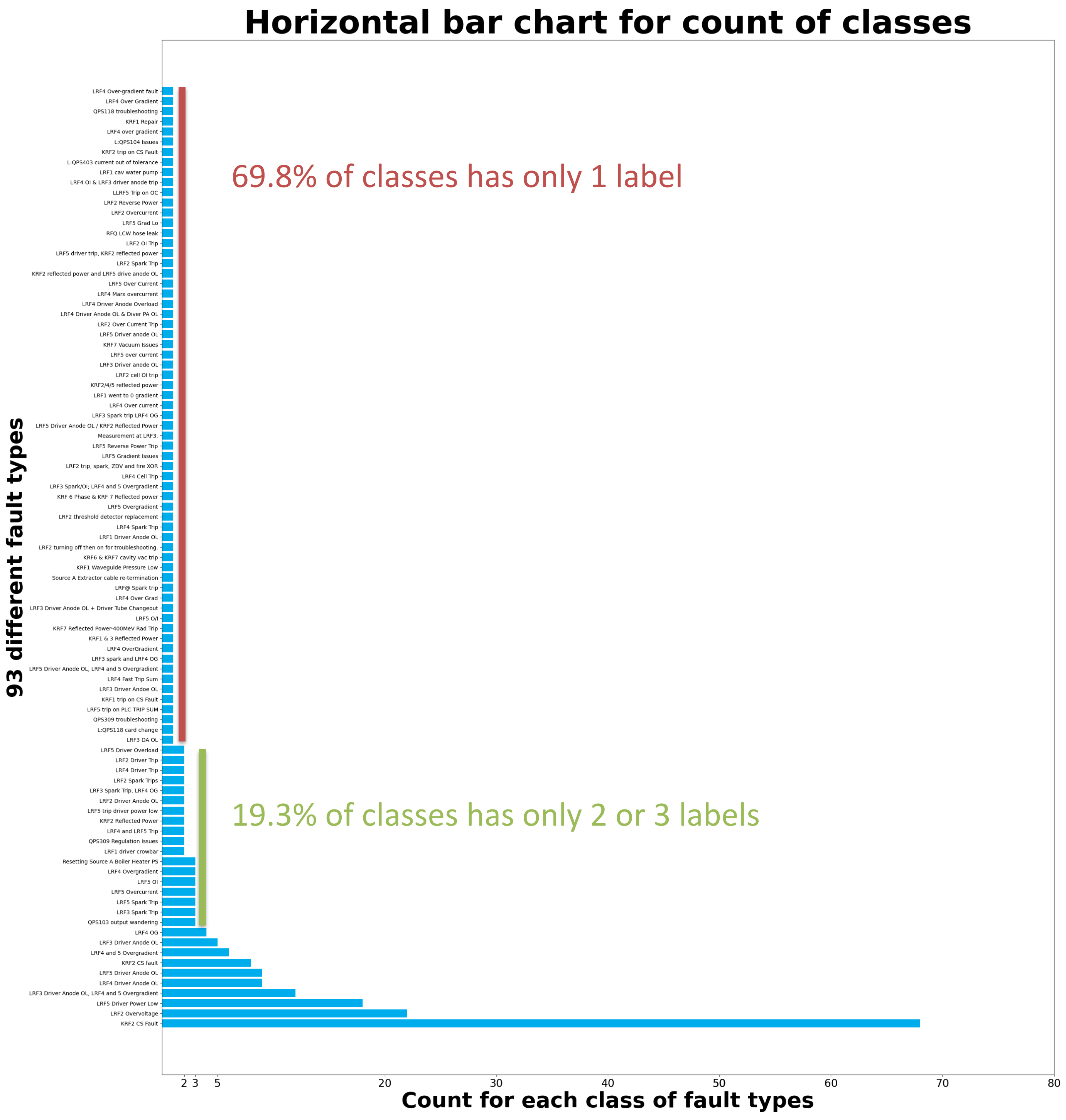

Figure 1 Plotting of the number of labels of each class.

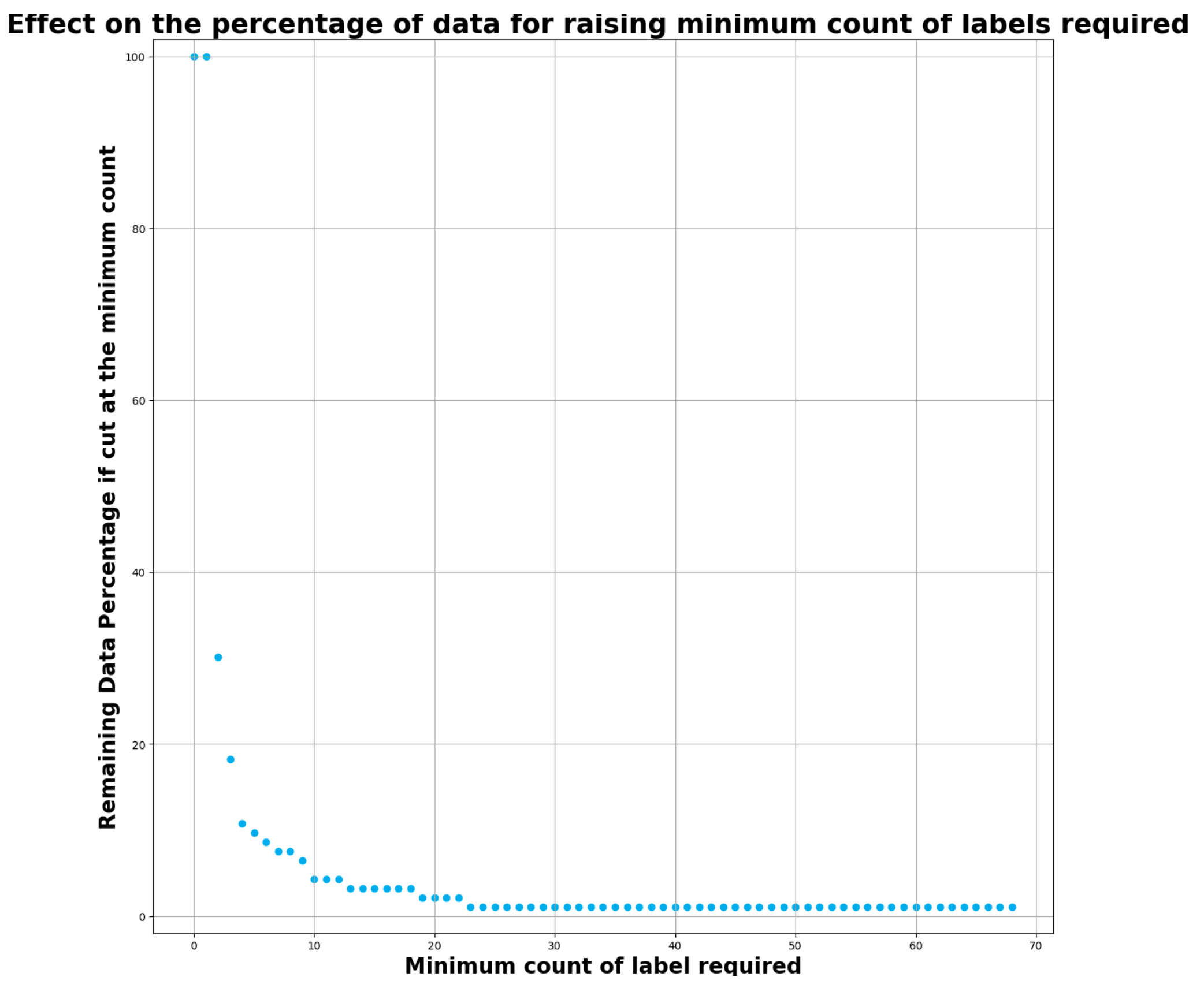

Figure 2 Dataset reduction percentage: measures the percentage that is being lost after applying minimum count threshold to the dataset (shortage of labels)

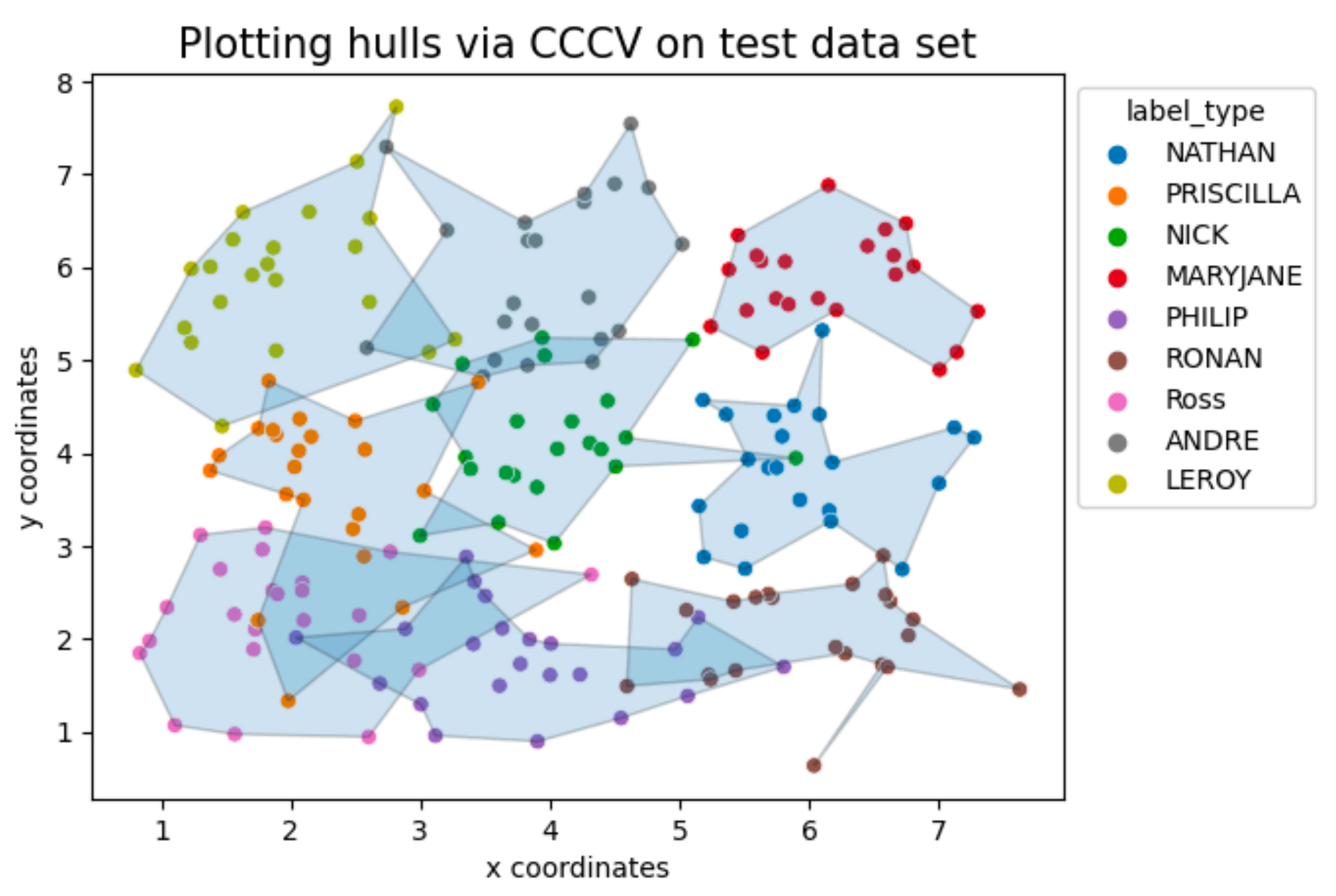

Figure 4 For each class: assigned a polygon that describes the region of classification.

Custom Convex/Concave Hull library

Made a library for understanding UMAP low-dimensional output. This 2D clustering library accepts a Pandas DataFrame that contains feature X and target y

- **Figure 4** For each class: assigned a polygon that describes the region of classification.
- For a new data point that is transformed through UMAP's space, check which clusters it is in and return a list of possible classes.
- **Figure 5** Visualizes a matrix of hull overlaps, quantifying how two classes might be related.

Development of (Label agnostic) clustering metric

The project has more unlabeled data available than it has labeled data, so we want performance metrics for both.

•Labeled metric development

Homogeneity: measures labeled similarity among data points in each cluster. [0, 1]

Completeness: measures if all data points of the same class are in the same cluster. [0, 1]

Minimum count threshold: a cut on the minimum label of a class.

We would want high homogeneity since potentially the clustering algorithms might cluster a class into multiple clusters, potentially the improving labeling.

•Unlabeled metric development

Silhouette: A measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). This score is bounded between -1 for incorrect clustering and +1 for highly dense clustering. Scores around zero indicate overlapping clusters.

> -> For comparison to labeled data metrics, shift up 1, divided by 2, to shift to the range [0, 1]. This helps with integrating into a uniformly weighted metric.

Conclusion / Future

- Activity now shifts focus to data acquisition, and continued improvements of these script.
- CI/CD (continuous integration / continuous development): Repo is large, requires hard thinking to integrate all files/work flow.

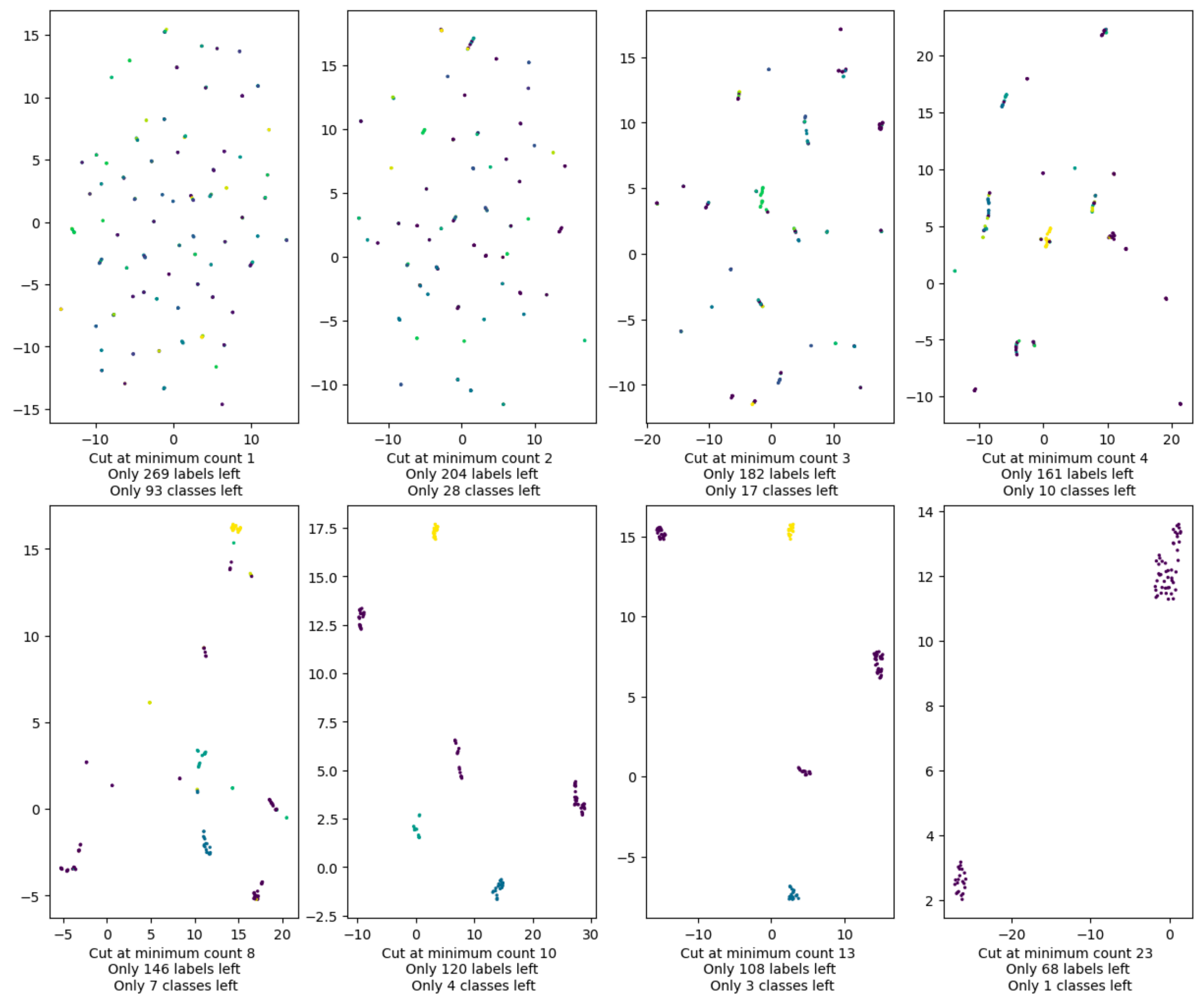

Figure 3 Change in the quality of clustering of UMAP(side effect) reflected via the minimum count with constant `min_dist` and `n_components` with data from /home/lcape/data/snappyfiles/outageFileCSV.csv

"`min_dist`" : 0, "`n_neighbors`" : minimum_count_of_config, "`n_components`" : 2

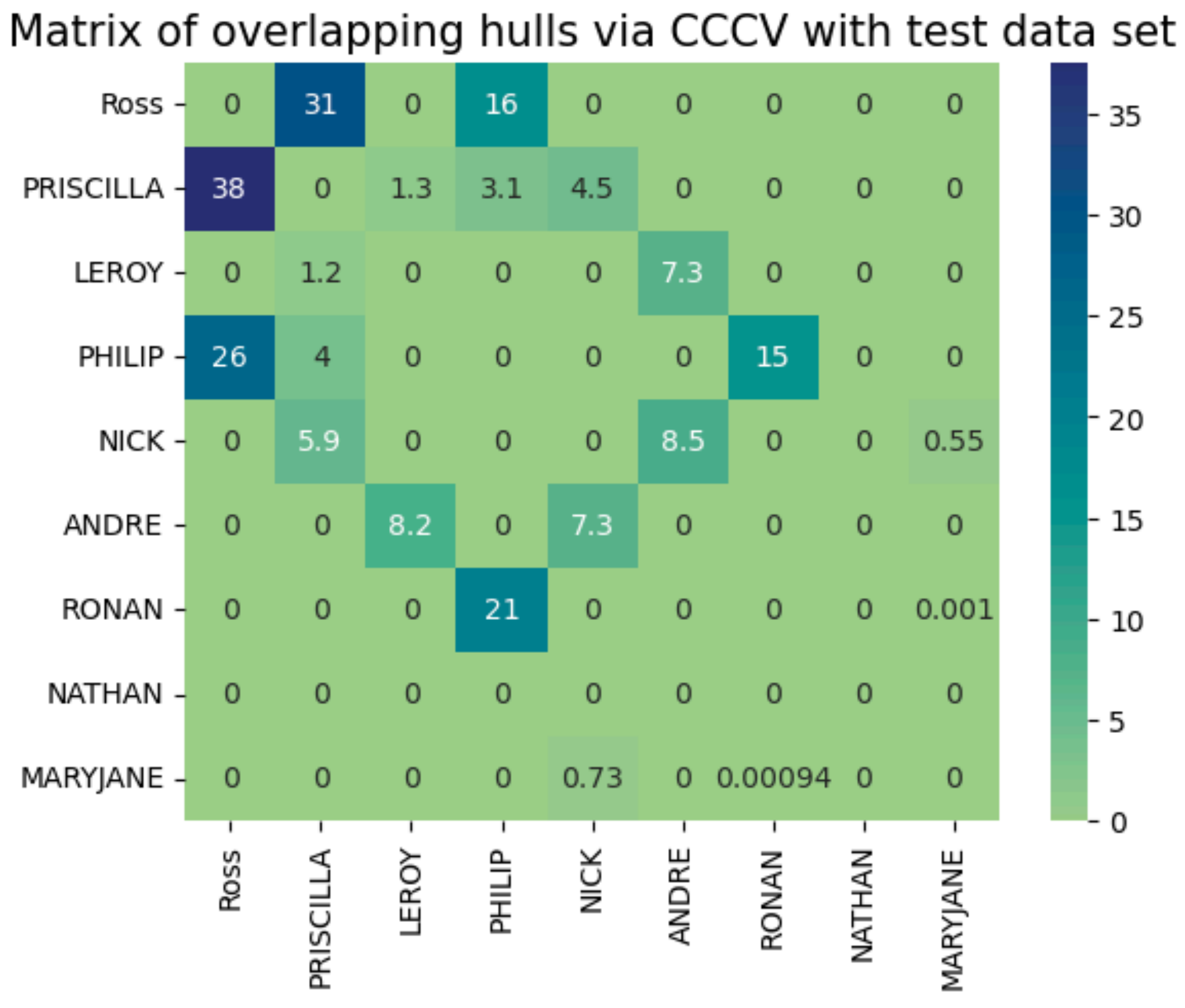

Figure 5: when reading the grid, if at the horizontal position "PHILIP", the vertical position "RONAN" is 30%, it means that the intersection of "PHILIP" and "RONAN" takes up 21% of the area of "PHILIP"