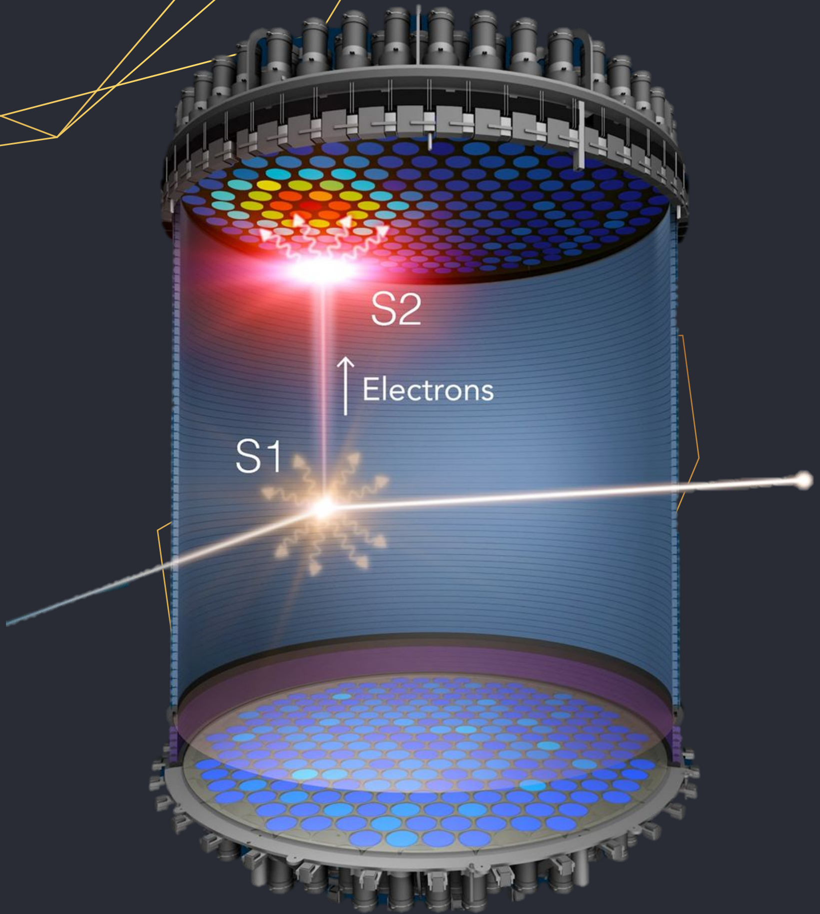
CLASSIFICATION OF PULSES IN THE LUX-ZEPLIN DARK MATTER DETECTOR

Elisa Ghetti

Kai Jenkins



LUX-ZEPLIN DETECTOR

Direct detection of dark matter based on liquid xenon scintillator

The interaction of an incident particle produces two signals:

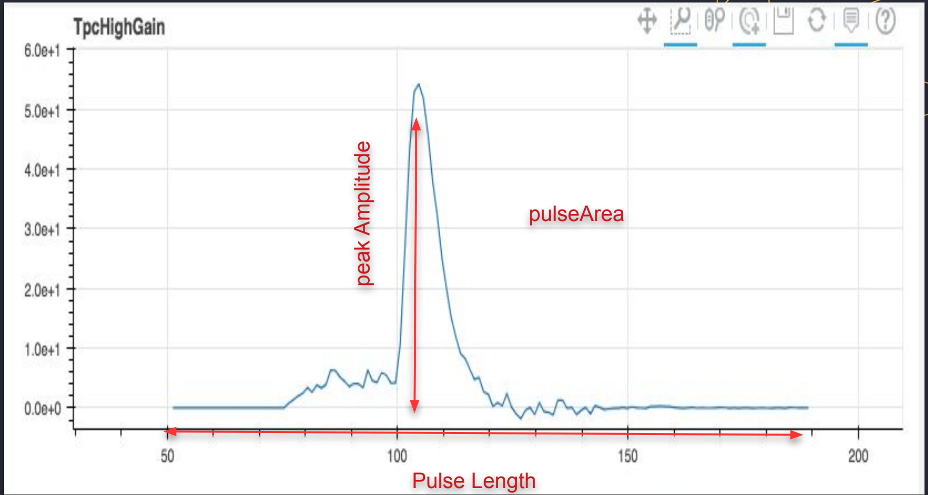
- **S1:** scintillation light
- **S2 :** electroluminescence light

Machine learning algorithms can be used to discriminate between them.

Goal: reach **>99%** overall classification accuracy

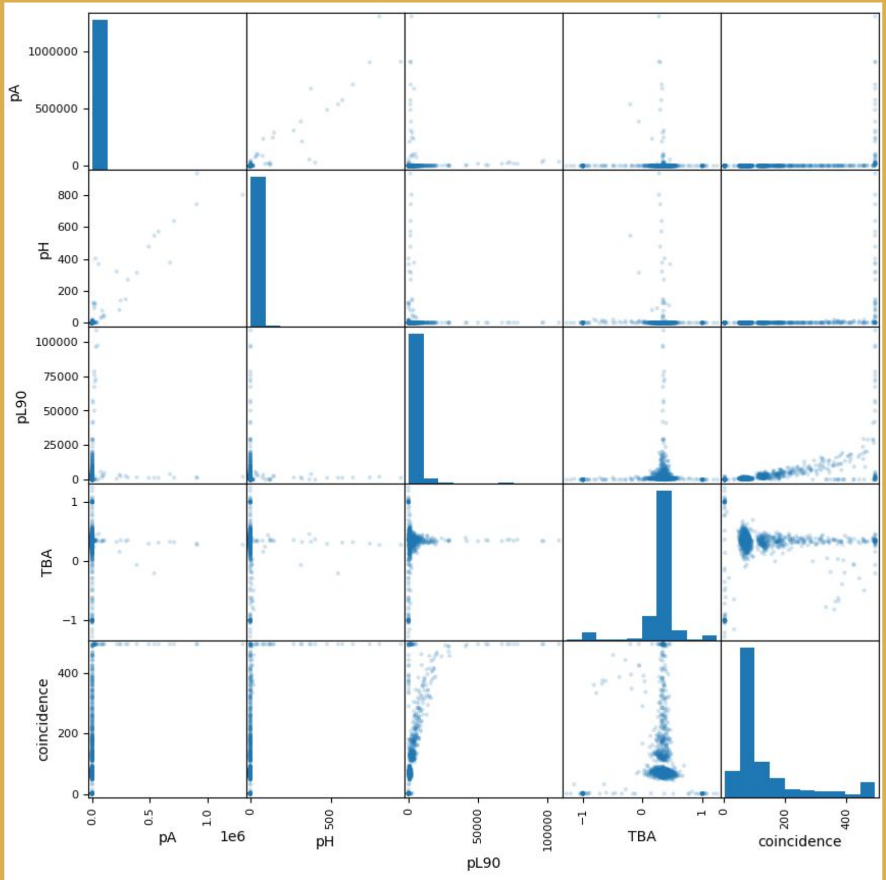
Classifier Input - 20 features (RQs):

- Pulse area (**pA**)
- Pulse amplitude (**pH**)
- Pulse length (**pL, pL90** - length at 90% area)
- Prompt fraction (**pF**) - fraction of area at start of pulse: 50, 100, 200, 500, 1k, 2k and 5k ns window
- Top-bottom asymmetry (**TBA**)
- Area fraction time (**aft**) time when pulse reaches X% of total area: 5%, 25%, 50%, 75%, 95%
- Peak Time (**pHT**) Time of maximum
- RMS Width (**pRMSW**)
- (**coincidence**) Number of channels that had non-zero contribution to pulse



Classifier Output - 4 classes:

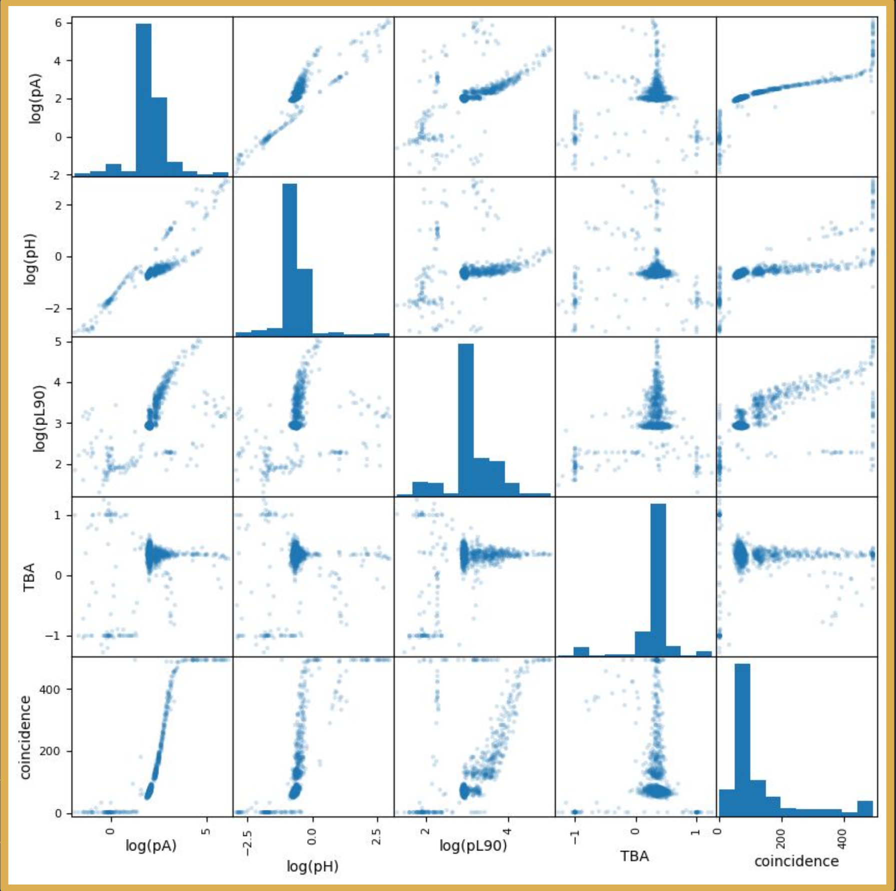
[0] Other
[1] S1 (scintillation)
[2] S2 (electroluminescence)
[3] SE (single electron)



FEATURE RESCALING

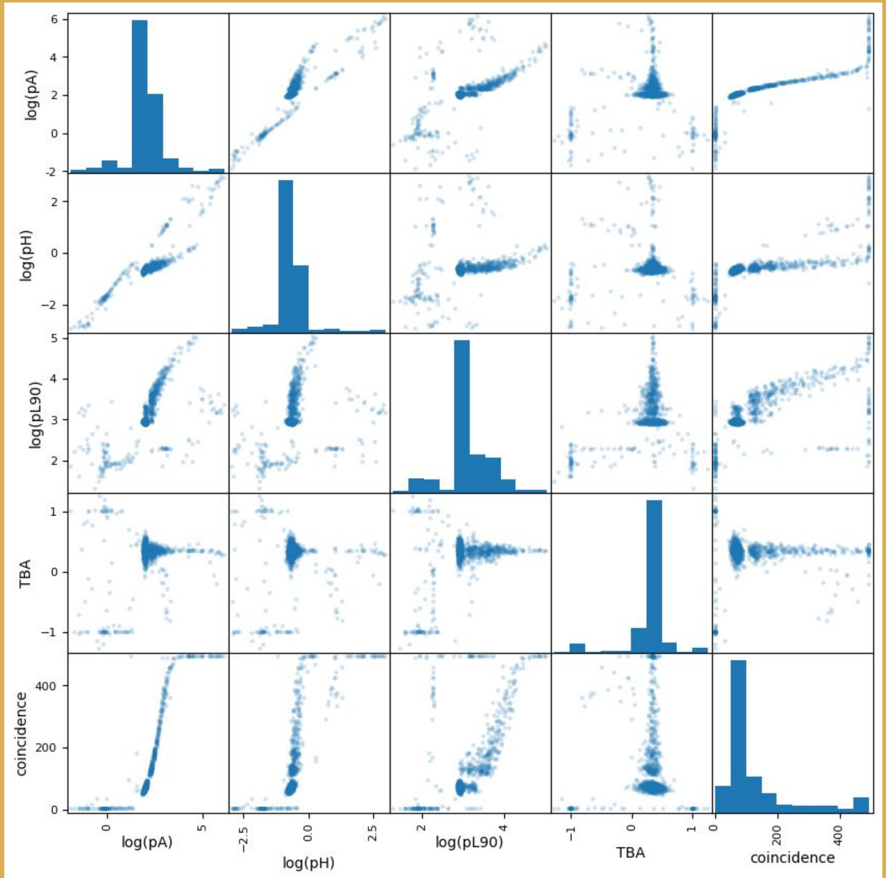
Data has to be in similar scale to avoid domination of features with larger values

FEATURE RESCALING



Data has to be in similar scale to avoid domination of features with larger values.

pA → log(pA)
pH → log (pH)
pL90 → log(pL90)



FEATURE RESCALING

Data has to be in similar scale to avoid domination of features with larger values.

pA → log(pA)
pH → log (pH)
pL90 → log(pL90)

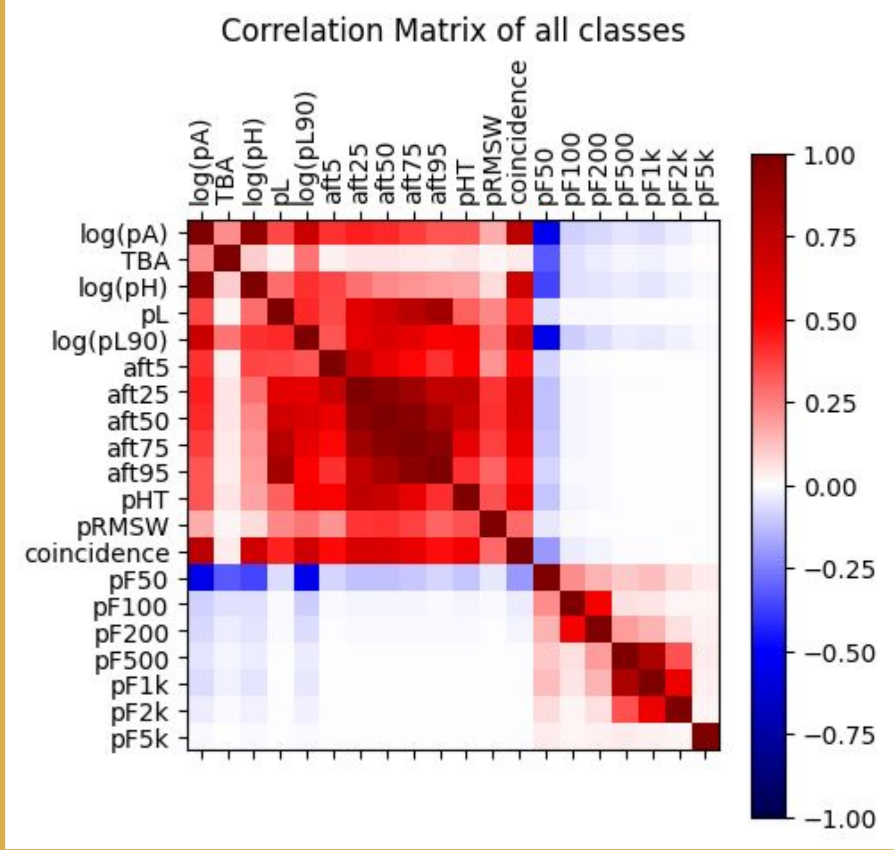
NORMALISATION

StandardScaler normalisation:

- Mean = 0
- Standard deviation = 1

CORRELATION MATRICES

Analysis of the **correlation** between features

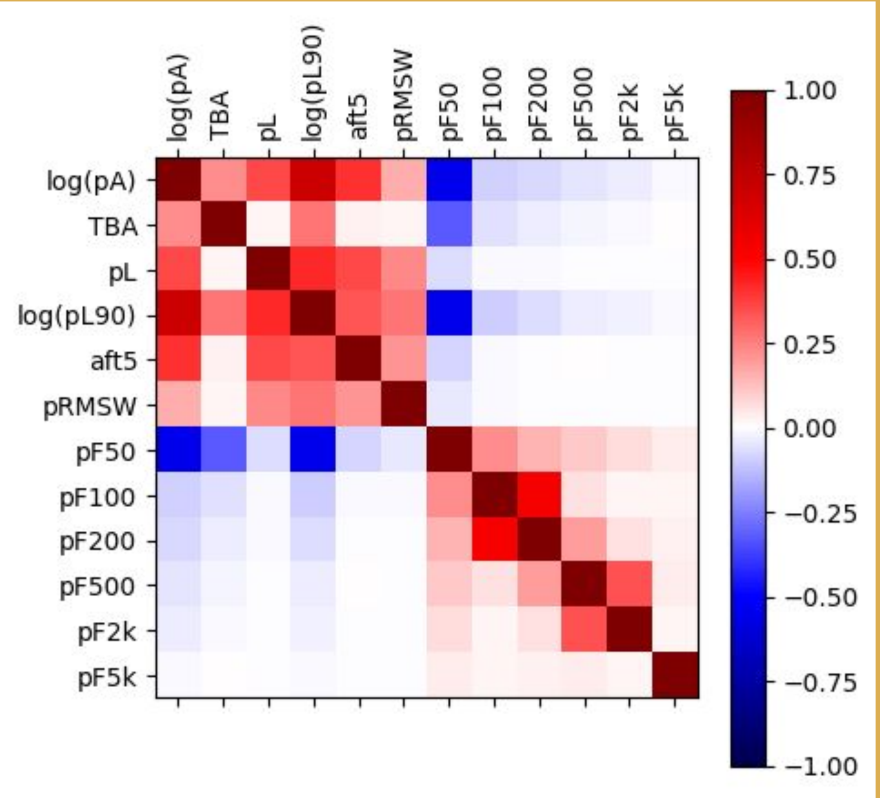


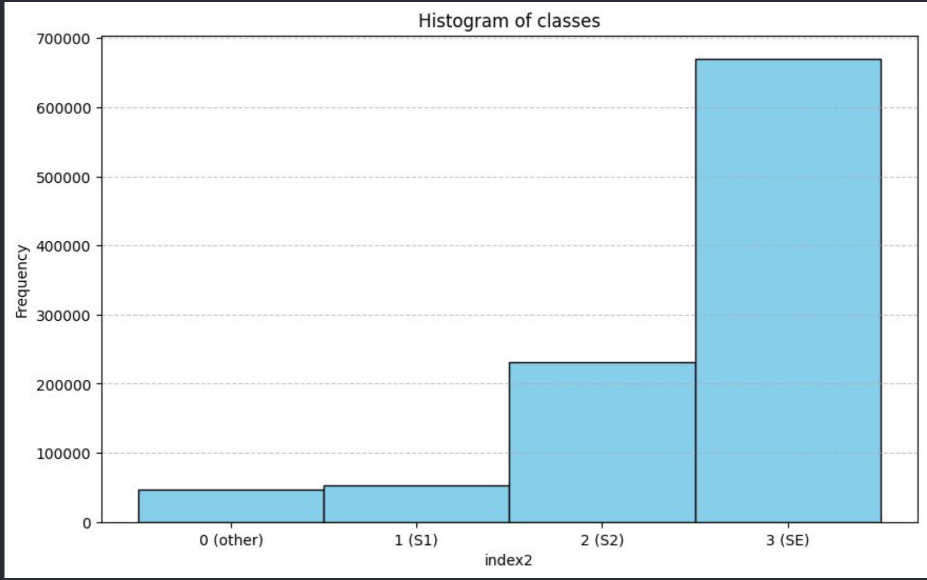
CORRELATION MATRICES

Analysis of the **correlation** between features

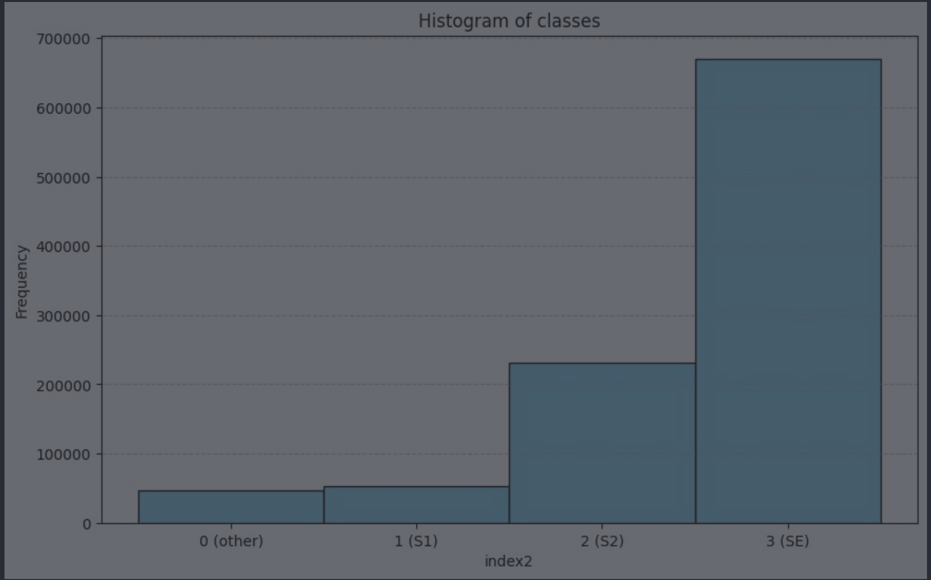
Highly correlated data can be rejected (adds no new information):

- log (pH)
- aft25, aft50, aft75, aft95
- pHT
- coincidence
- pF1k

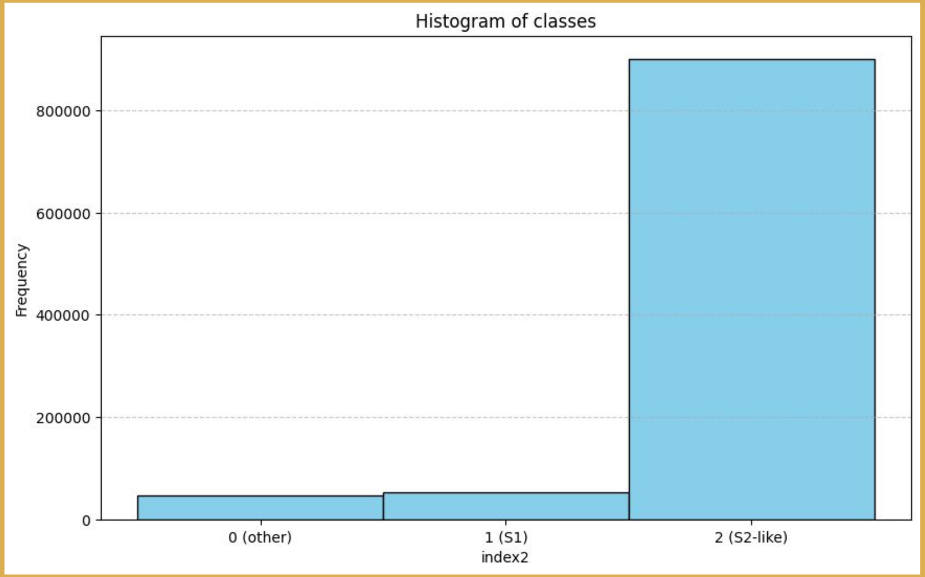




PREPROCESSING OF THE LABELS DATASET



PREPROCESSING OF THE LABELS DATASET



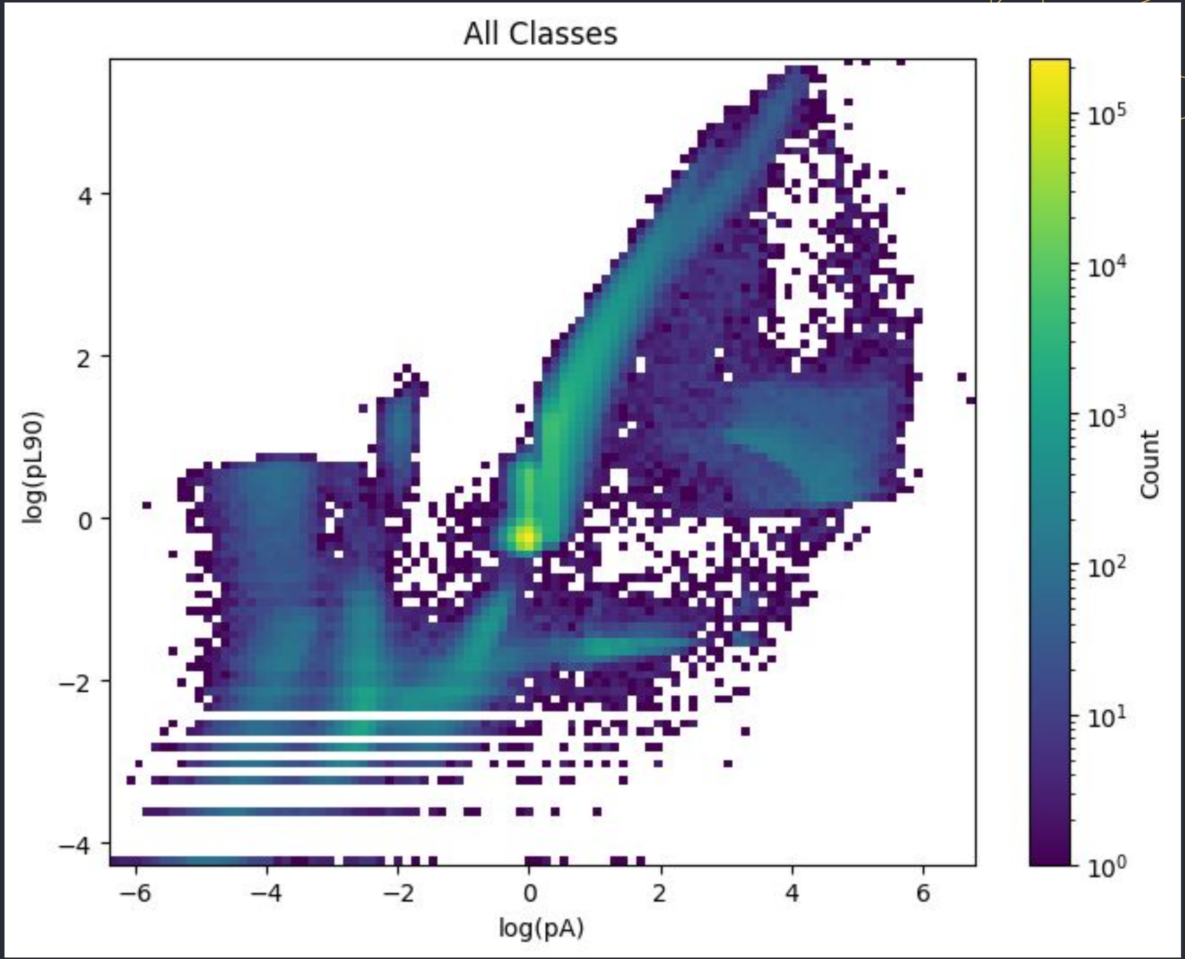
Labels S2 and SE can be combined into one S2-like label
(both produced by electrons)

Note: will be applying balancing in all our models to adjusting for S2 frequency

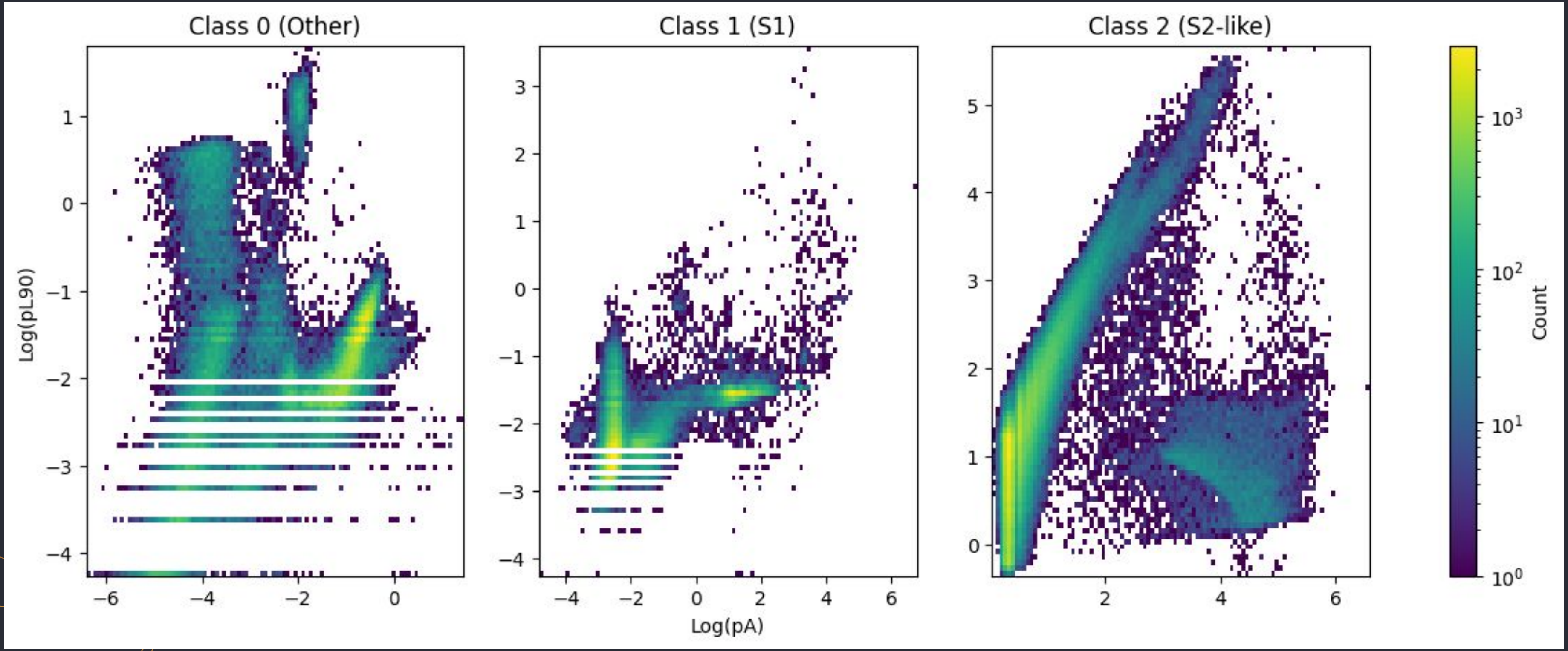
DENSITY PLOTS

- Looked at density plots between different features to visualise their relationships

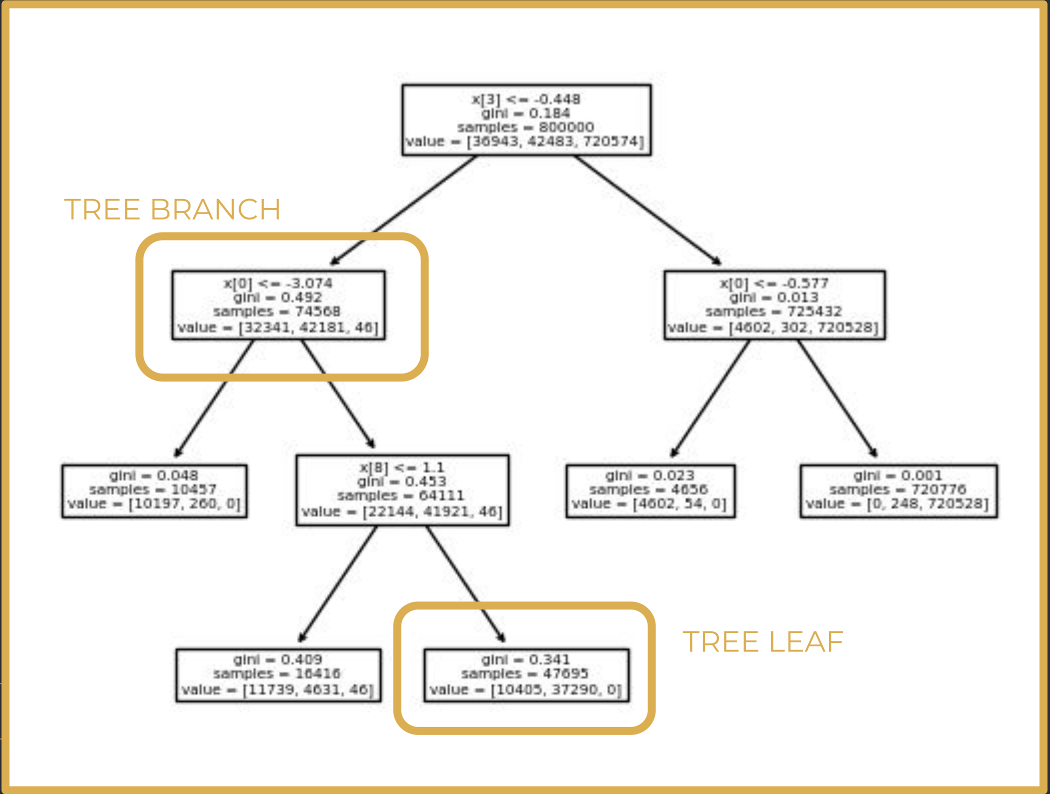
- Can already see some groups in this plot



DENSITY PLOTS (Individual classes)



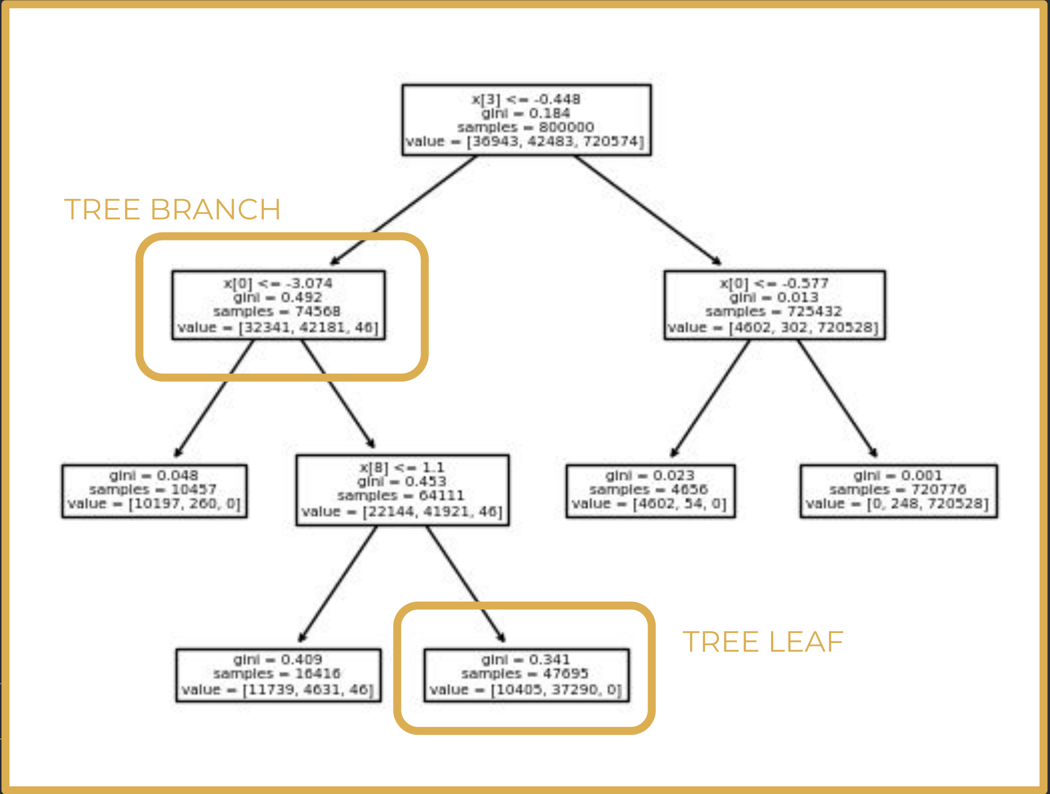
DecisionTreeClassifier MODEL



Recursive partitioning of the data based on the minimization of an impurity function

➔ **GINI** impurity (likelihood of new data being misclassified if given a random class label.)

DecisionTreeClassifier MODEL



PARAMETERS:

- *random_state*: set to 0 for reproducibility
- *max_leaf_nodes*
- *max_depth*

NOTE:
all hyperparameters in this project where optimized with OPTUNA

DecisionTreeClassifier MODEL

The model's performance can be tested by calculating the **score**:

→ Test set score: **98.80%**

A simple tree model is very simple yet powerful for a classification problem like this.

CONFUSION MATRIX

PREDICTED CLASS

CLASS LABEL	0	1	2
0	8358	830	3
1	909	9661	45
2	21	595	179579

RandomForest MODEL

- Ensemble of *DecisionTrees* where output is selected by **majority vote**

- **Bootstrapping:**

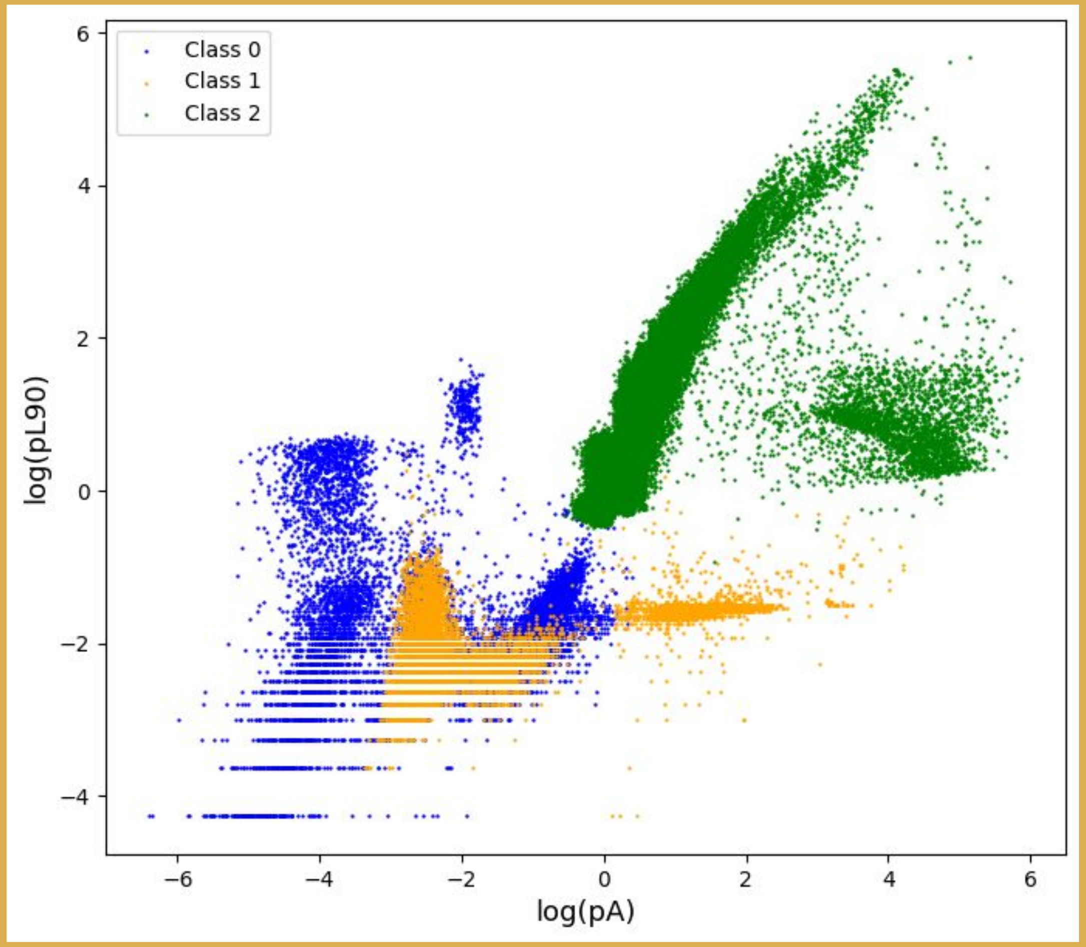
  → Reduces bias
  → More resistant to **overfitting**

RandomForest MODEL

Test set score: **99.13 %**

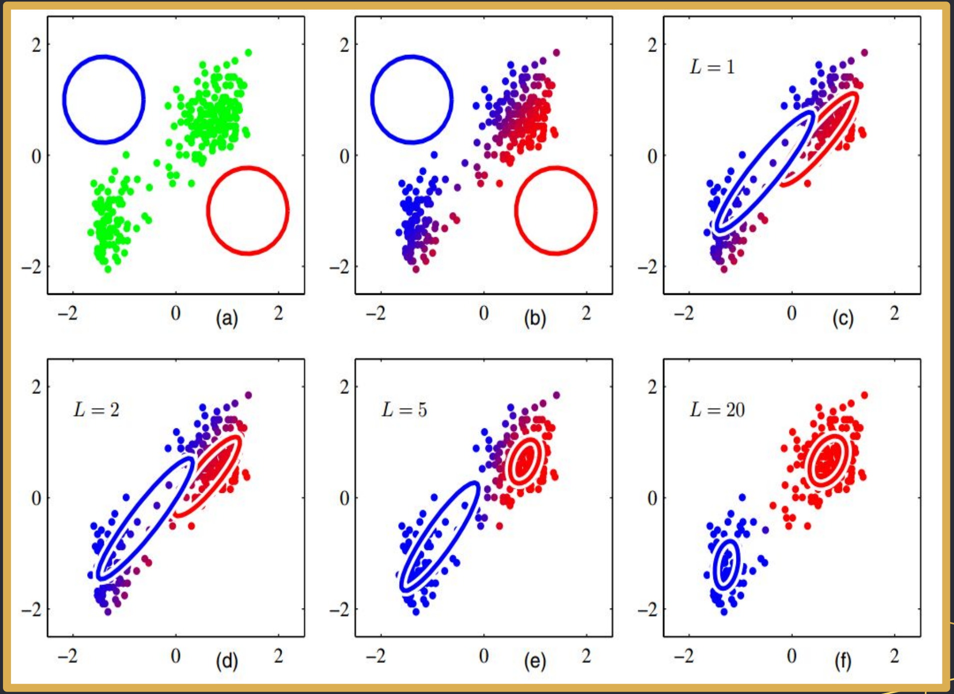
CONFUSION MATRIX

CLASS LABEL \ PREDICTED CLASS	0	1	2
0	8271	920	0
1	732	9844	39
2	44	0	180150

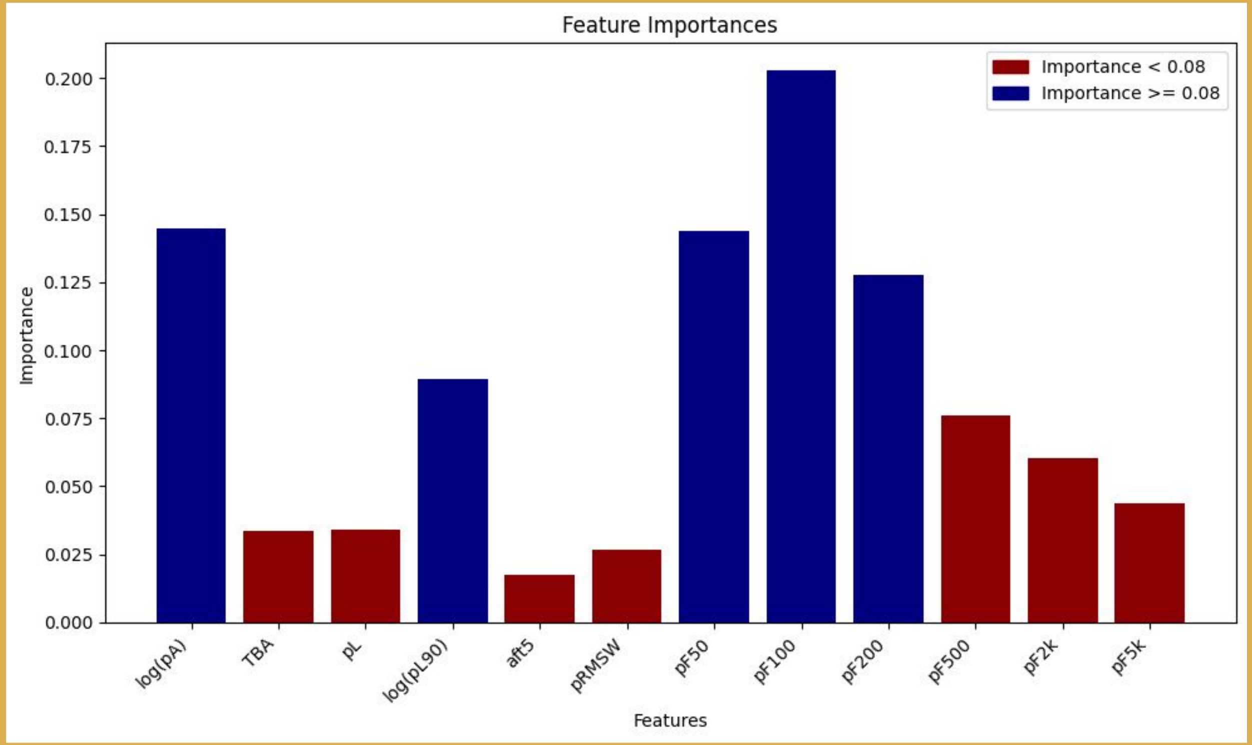


GaussianMixture MODEL

- **Unsupervised** learning
- **Clustering analysis**: data is assumed to be distributed in a finite number of clusters
  - ➔ Linear superposition of **K gaussian distributions**



FEATURE IMPORTANCE



Relative importance of each feature:

how much the tree nodes that use that feature reduce impurity on average

GaussianMixture MODEL

GMM is a **density based** algorithm: a large number of components is necessary to fit less dense regions of the data

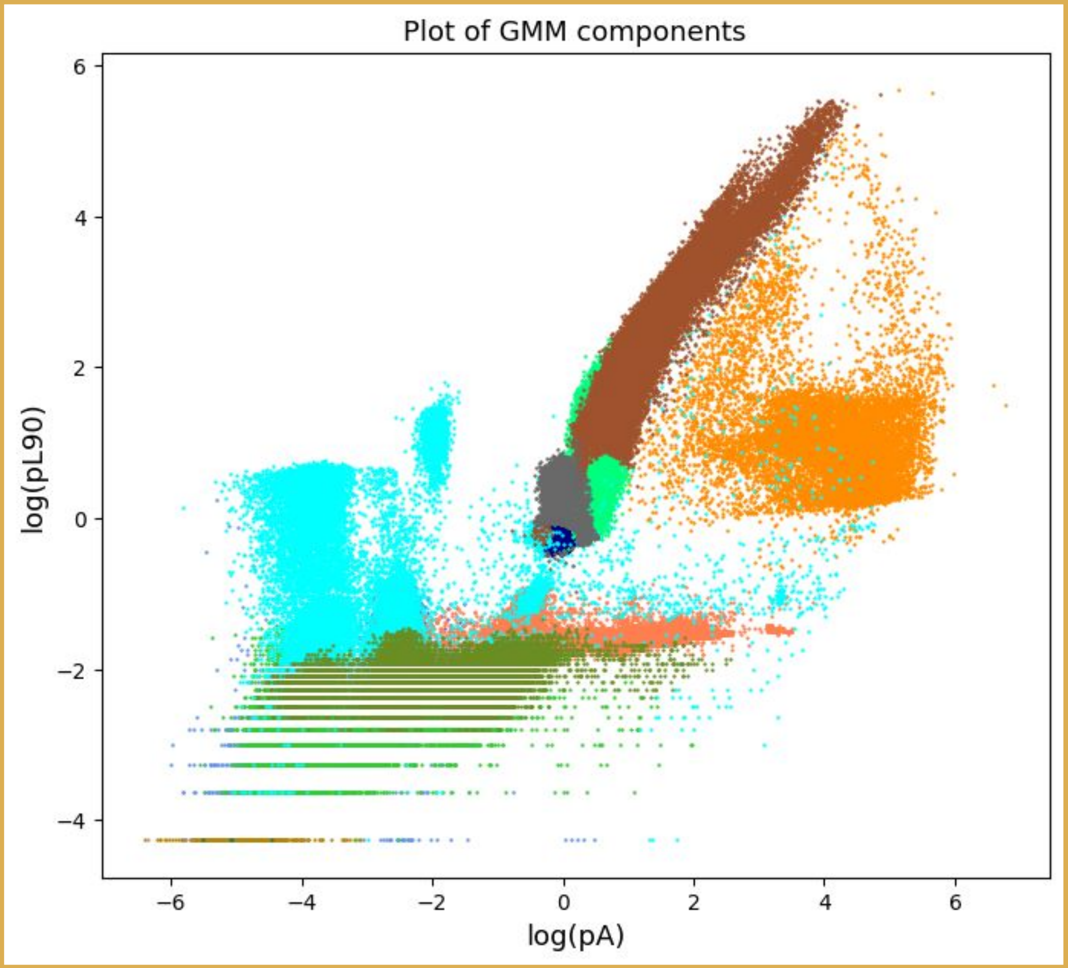
→ K has to be much larger than the number of classes

GaussianMixture MODEL

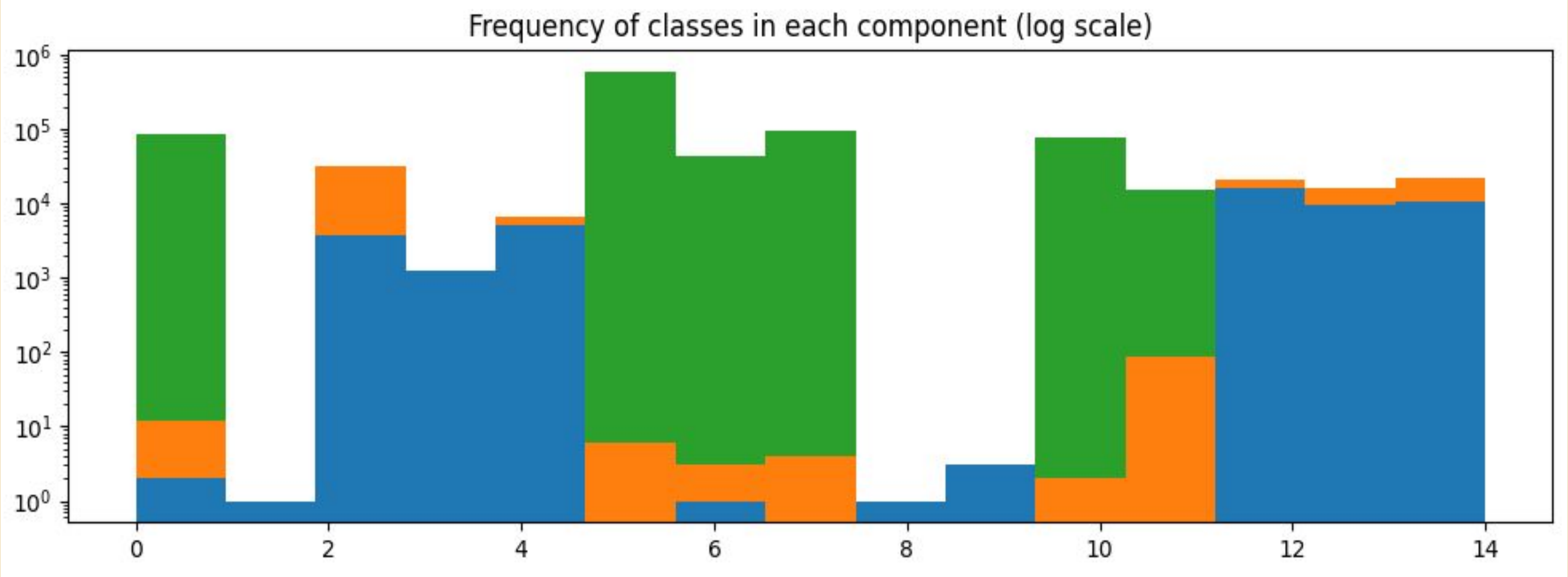
GMM is a **density based** algorithm: a large number of components is necessary to fit less dense regions of the data

→ K has to be much larger than the number of classes

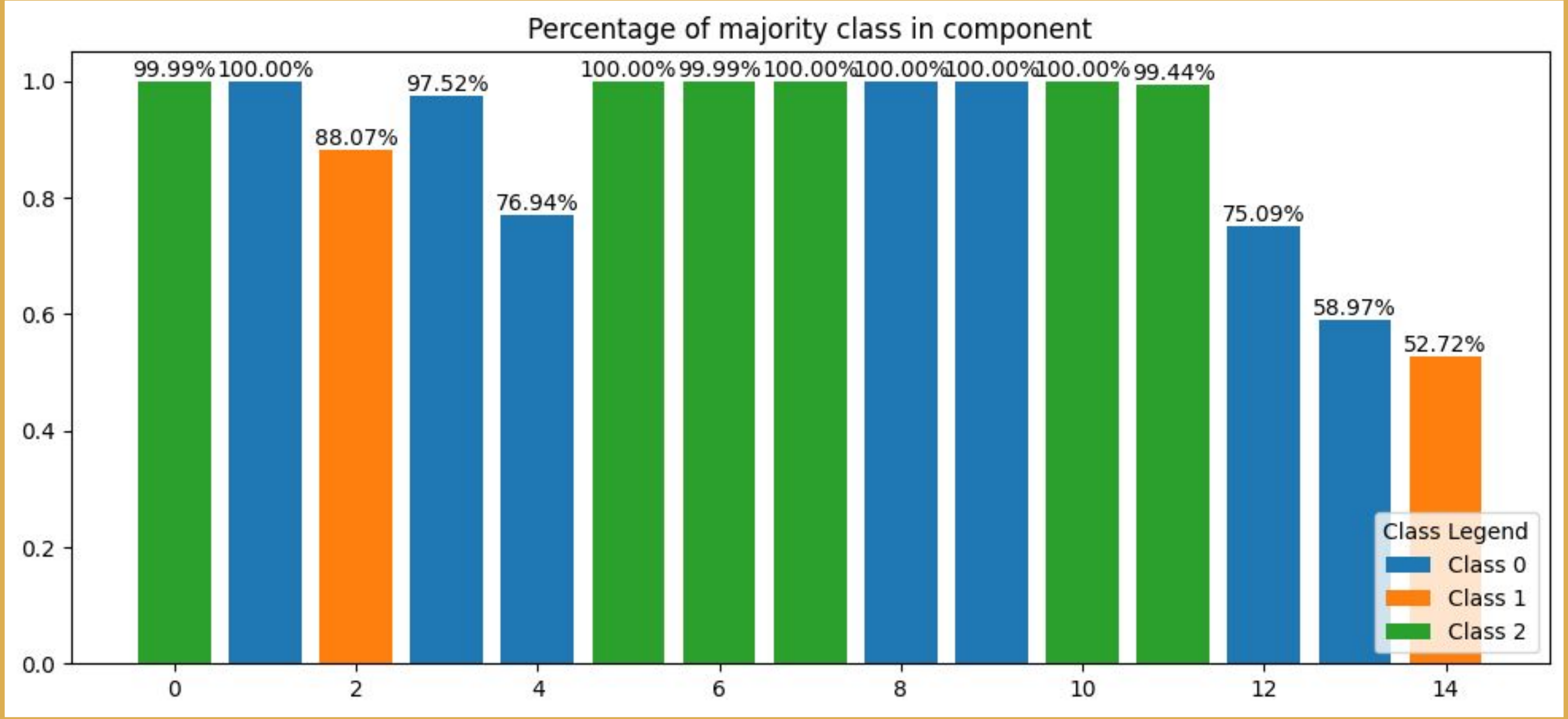
Fit with **K=15** gaussians

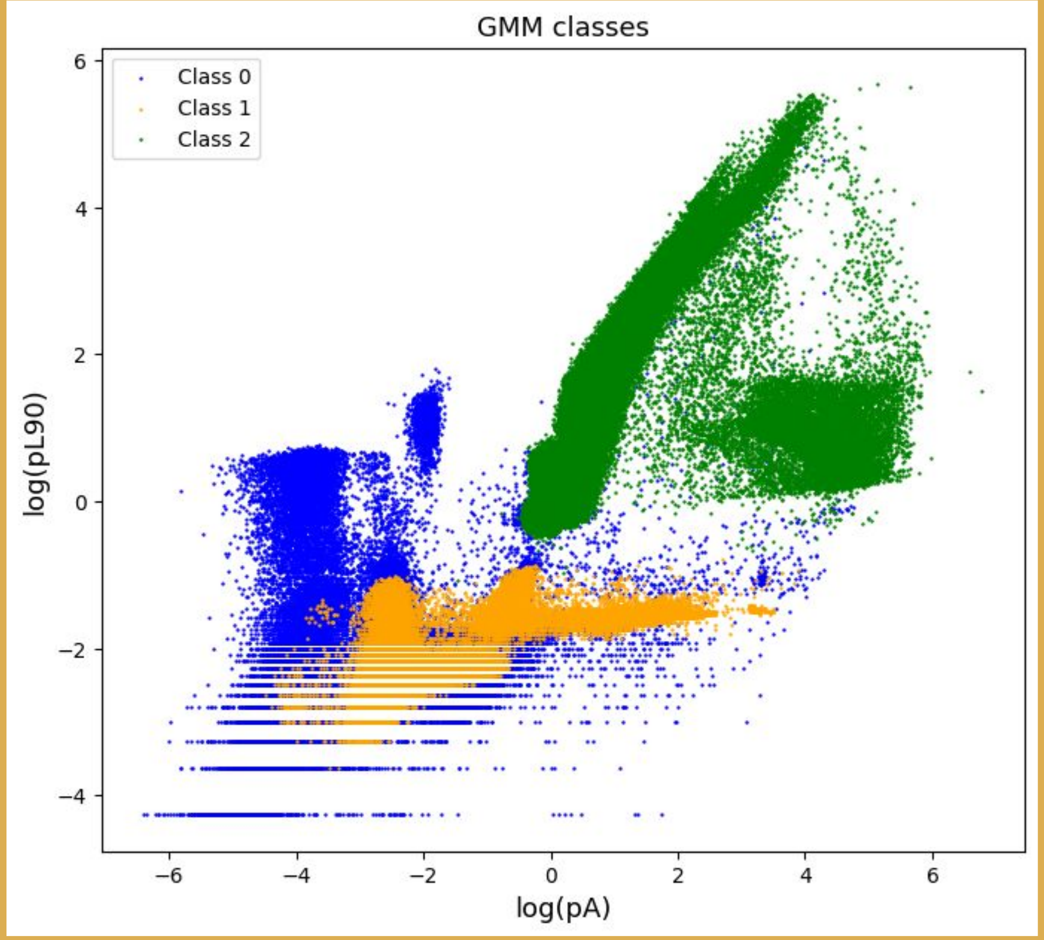


GaussianMixture MODEL



GaussianMixture MODEL





Each gaussian component is associated to its majority class

New labels dataset can be used to train a more accurate Forest model

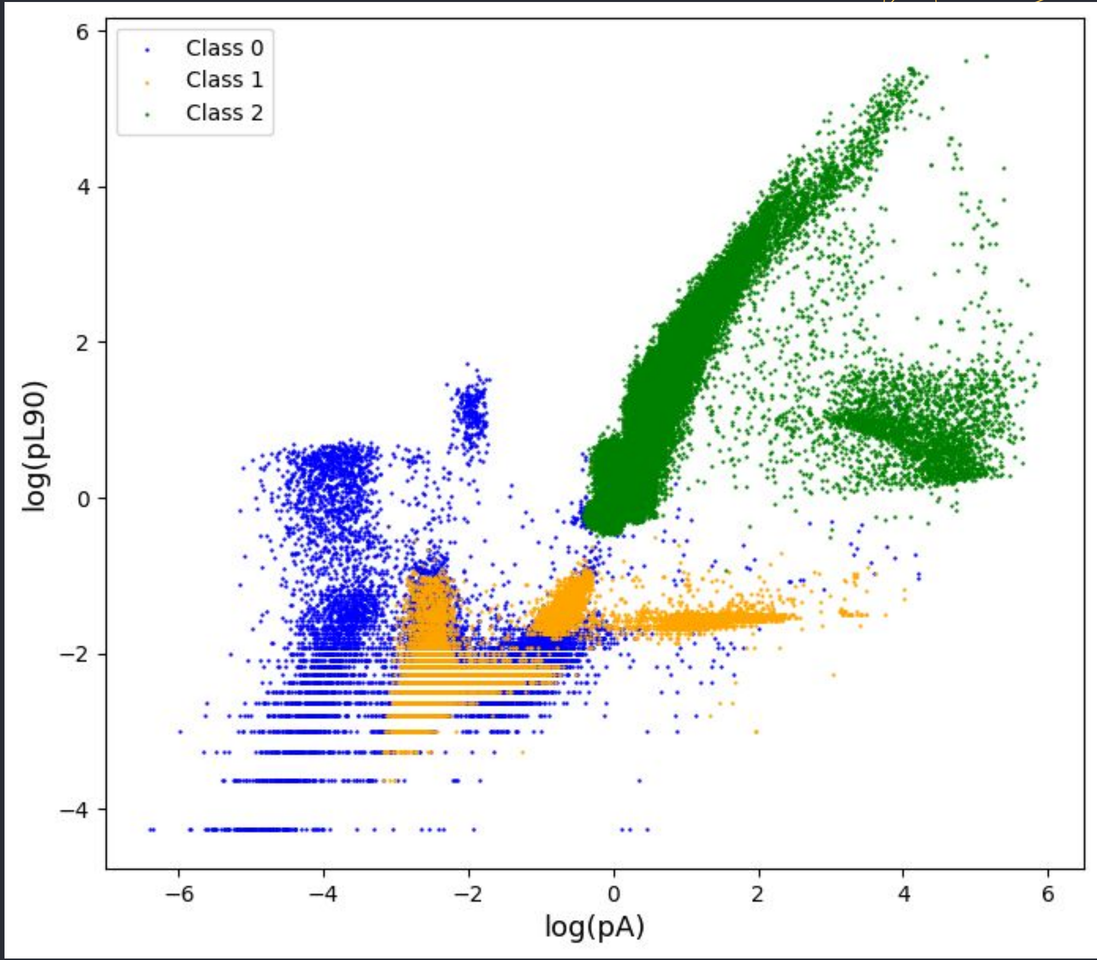
GMM *RandomForest*

Testing score: **99.40 %**

CONFUSION MATRIX

PREDICTED CLASS

CLASS LABEL	0	1	2
0	8377	555	39
1	492	10370	0
2	122	0	180045



CONCLUSIONS

DECISION TREE

Score:
98.80 %

CLASS LABEL \ PREDICTED CLASS	0	1	2
0	89.9%	7.49%	0.002%
1	9.7%	87.2%	0.03%
2	0.2%	5.4%	99.97%

RANDOM FOREST

Score:
99.13 %

CLASS LABEL \ PREDICTED CLASS	0	1	2
0	88.8%	9.38%	0%
1	9.99%	90.6%	0.02%
2	1.23%	0%	99.98%

RANDOM FOREST WITH GMM DATA

Score:
99.40 %

CLASS LABEL \ PREDICTED CLASS	0	1	2
0	93.2%	5.08%	0.022%
1	5.47%	94.9%	0%
2	1.13%	0%	99.98%

* percentage of actual class label over total predictions of one class label

FUTURE DEVELOPMENTS

PERMUTATION IMPORTANCE

Randomly permuting variables in a tree and comparing its accuracy with the one of the original tree

→ accounts for **highly correlated** features

INCREASING K

Better fit of less dense regions and decrease in relevance of singularities

NEURAL NETWORK (TriNeT)

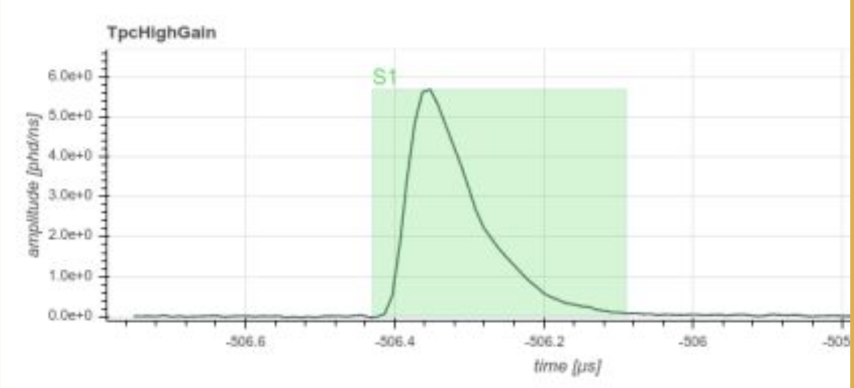
Ensemble of Neural Networks which focus on separating one feature from the others

THANK YOU FOR YOUR ATTENTION

BACKUP

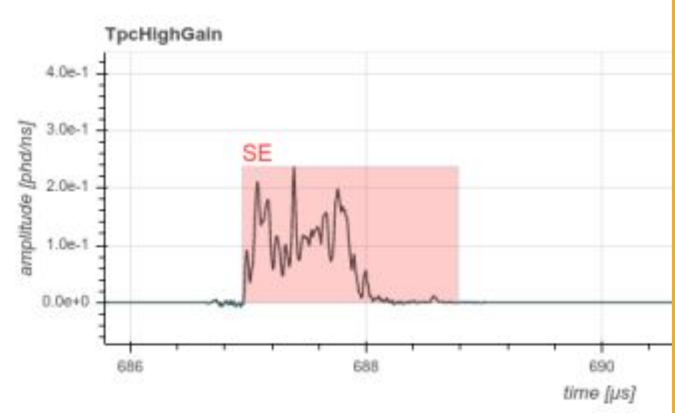
PULSES IN LZ

S1

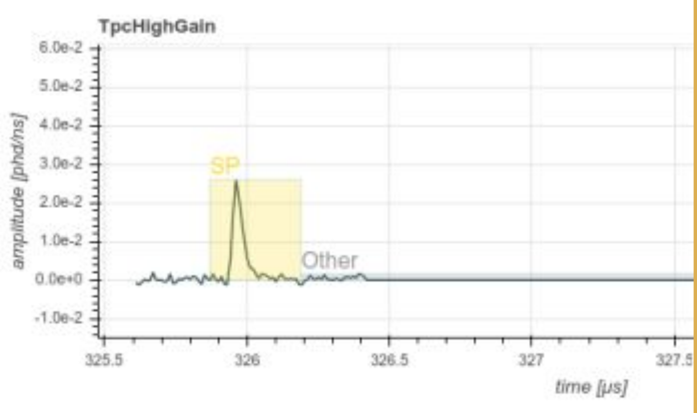


S2

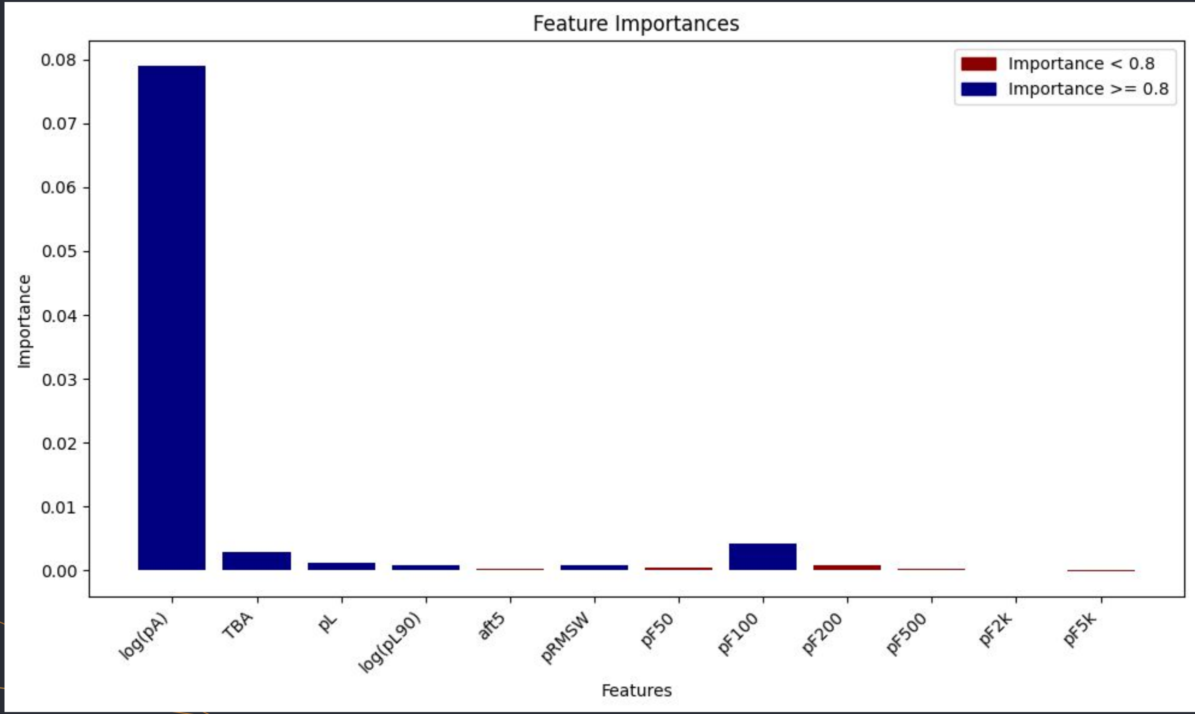
SE



OTHER



PERMUTATION IMPORTANCE

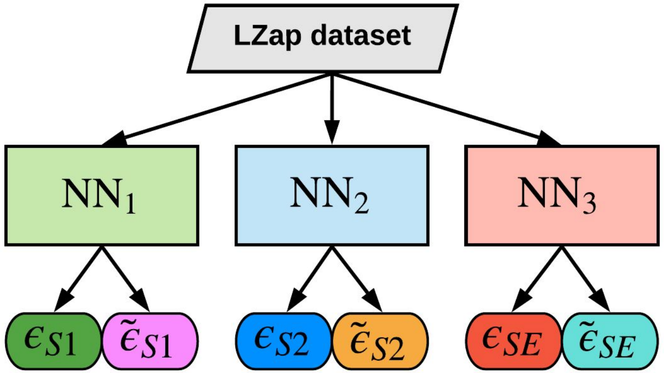


Randomly permuting variables in a tree reduces its efficiency

➔ Comparing its accuracy with the one of the original tree you can get the variable's importance

Permutation importance accounts for **highly correlated** features

TriNet CLASSIFIER

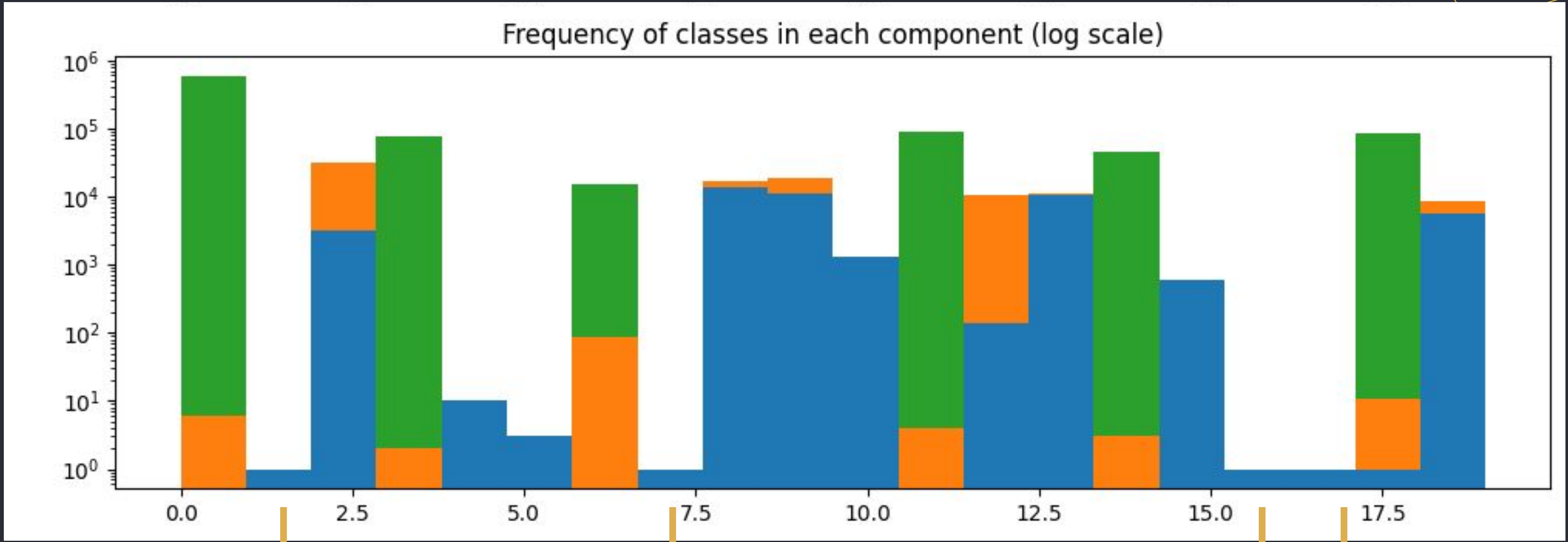


Ensemble of Neural Networks trained ad **One-VS-All**:

→ Each NN only learns one designated class, the rest of the pulses are labelled as “not of that class”

→ Trained using pre-existing labels dataset

GMM WITH K=20



GMM with 15 or 20 doesn't change much, as the number of singularities increases: to see improvement we would need a much larger K, which requires too much computational power