ML4NP: Meeting 03.07.2020

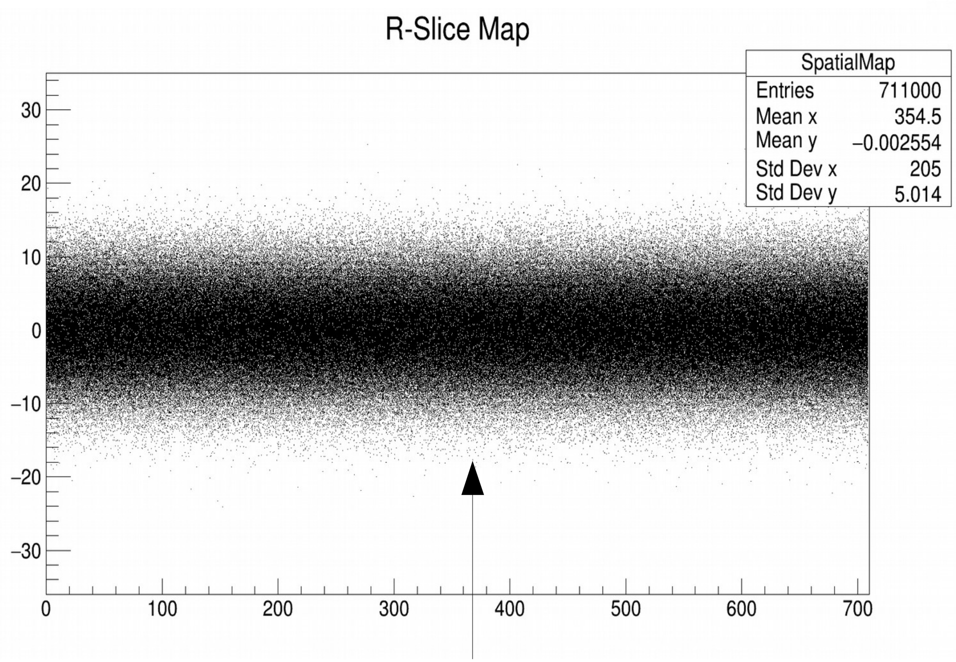
Summary

- Spatial Map:
  - recap
  - updates: inner/outer maps
  - questions

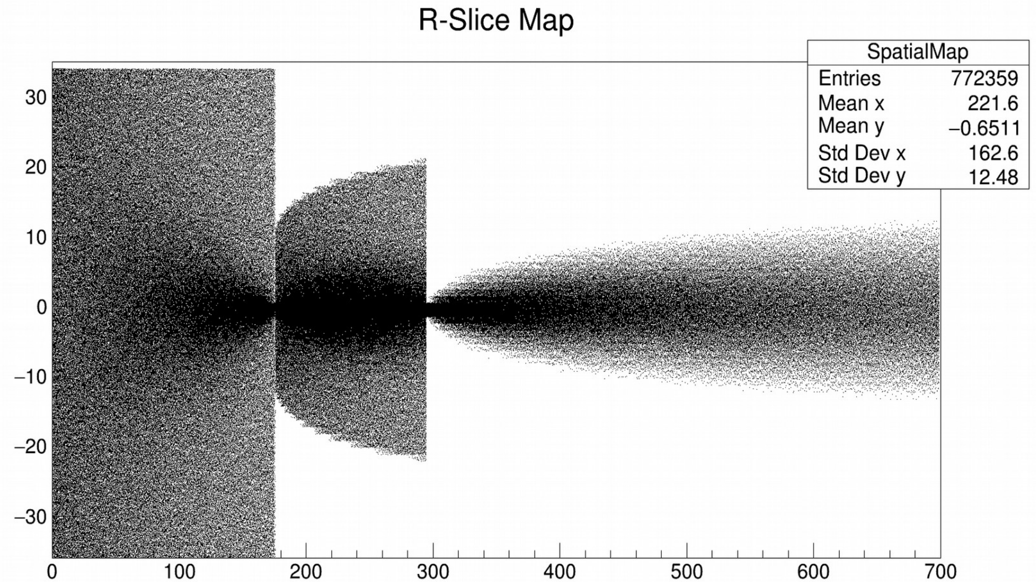
- Features for ML
- Classification wt DTree

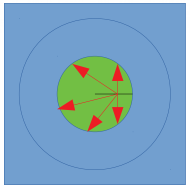
Spatial Map (recap)

- Sampling the PE hit wt Gaussian does not reflect the real setup.
  Problem: it doesn’t take into account the radius!
- **Idea**: with a bit of geometry and patience, we can compute the intersection of OP trajectories (*straight line*) with the shrouds (*circle in X-Y*)
- Is it real now? NO! Real Map should consider R, Z, angle.
  But it should be more realistic.

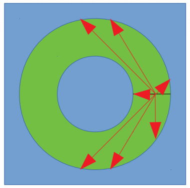


This is how the Gaussian hit-space distribution would look like as map

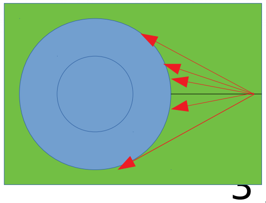




Only Inner



Inner/Outer

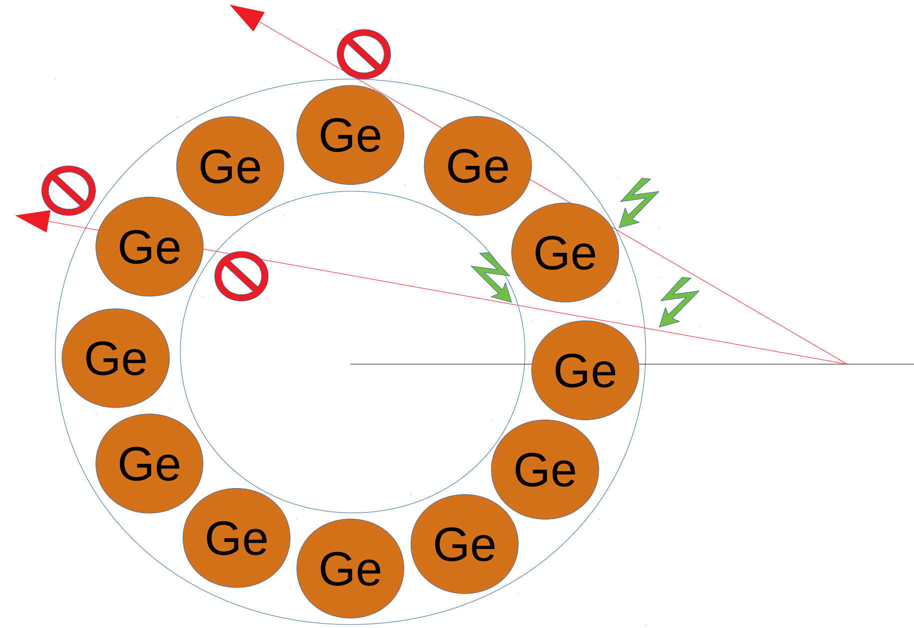


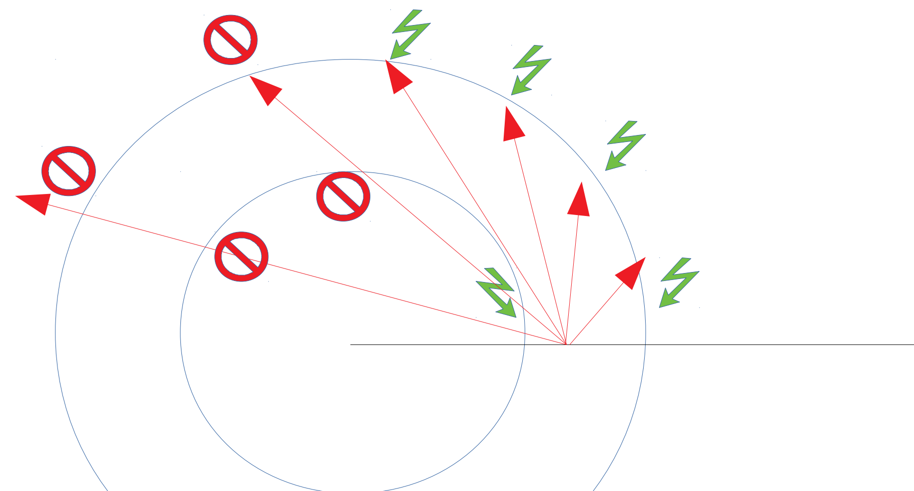
Only Outer

Spatial Map (updates)

- Last meeting’s observation: LGND200 has 2 shrouds. We should have Inner and Outer Detections!

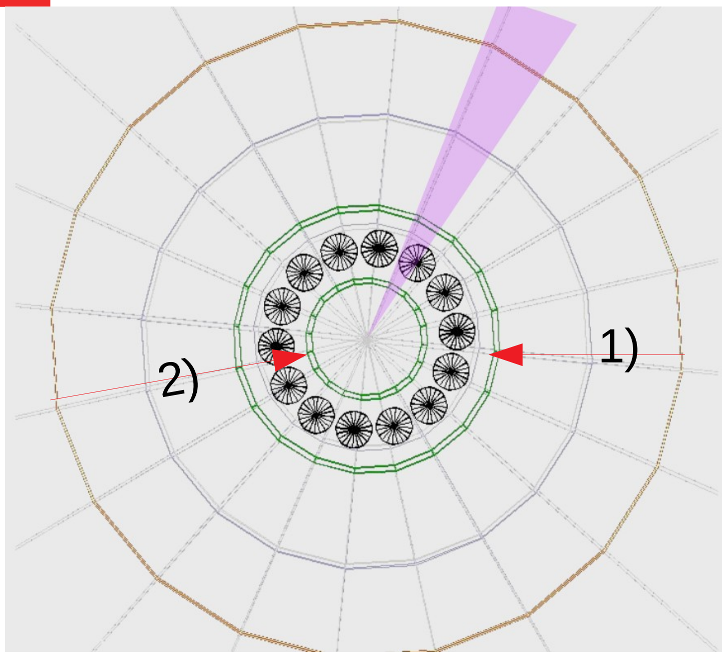
- **Updates**:
  - Creation of <u>3 Maps</u>:
    PrInnerDet, InnerMap, OuterMap
    Why 3? *See next slide*.
  - Integration of <u>other scenarios</u>:
    - Photon from outer region not captured by the outer shroud and detected by the inner one, and viceversa
    - How? Approximate with probability!





I don’t know how to treat the spread between Inner and Outer Shrouds

Spatial Map (updates)



Fibers don’t cover the whole cylinder surface.
Only the **54%** is covered (CJ email).

Ge Arrays could reflect the photons.
I guess only **10%** photons pass Ge,
90% are reflected.
Could we refine these values?

I assume **two scenarios** for a photons starting in the outer region:

1) Pr(**Hit Outer Shroud | Start in Outer Region**) =
   = Pr(captured by OuterShroud) + Pr(reflected by Ge) * Pr(captured by OuterShroud)
   = .54 + (1-.54) * .90 * .54 = .76356

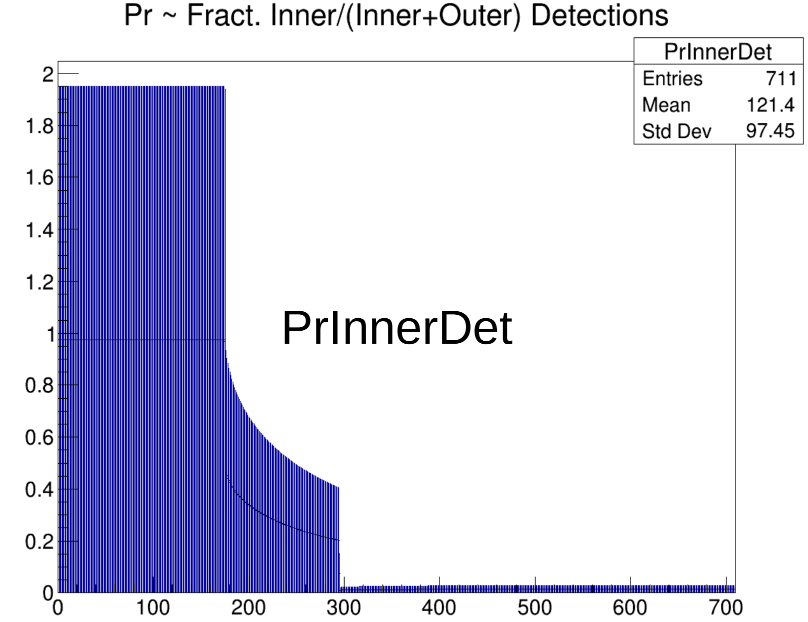
2)Pr(**Hit Inner Shroud | Start in Outer Region**) =
   = Pr(not captured by OuterShroud) * Pr(pass Ge) * Pr(captured by InnerShroud)
   = (1-.54) * .10 * .54 = .02484

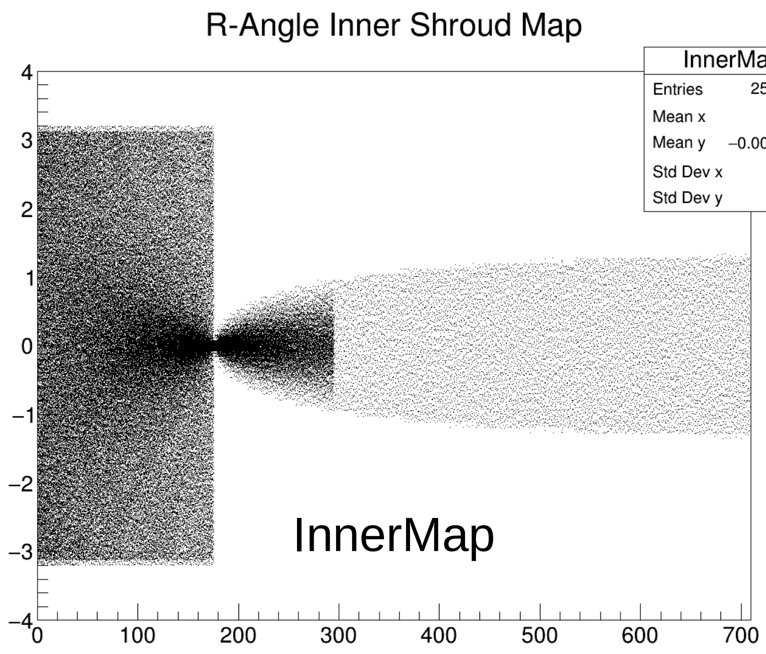
Since, we aim to distribute the detection, we normalize these probabilities:
1) 0.03151 2) 0.96849
...and use these two value as weights to fill the Spatial Map!

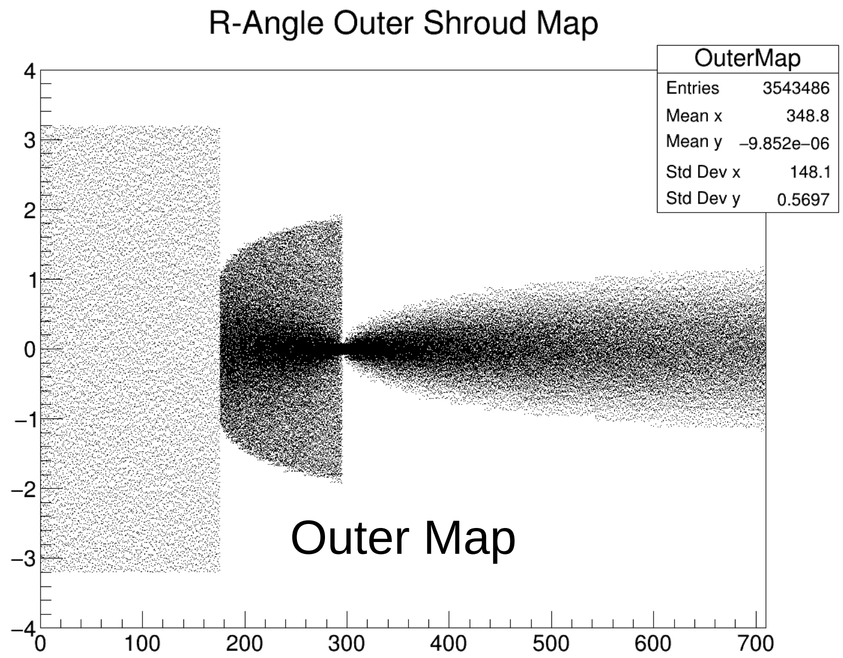
Spatial Map (updates)

- How to combine 2 maps?
  We have NPE = OPMap(Edep)

  1) Use PrInnerDet to compute *NPE_in* that hit the InnerShroud and *NPE_out* that hit the OuterShroud
  2) Spread NPE_in in InnerShroud wt InnerMap
  3) Spread NPE_out in OuterShroud wt OuterMap







Spatial Map (questions)

- All the features about Spatial Spread of detections depends on the map
- If the reality != simulation, the resulting “rejection strategy” would be uneffective
- Toy Spatial Map: how to deal with Ge? Tips?
- To have more realistic Spatial Map:
  It’s time to talk wt Patrick and Mario (TUM).
  - They already produce data wt symmetry
  - If their simulator is open, we can use it
  - Otherwise, we can ask them for raw data
  - We have to think to a map: R, Z, deposit-angle→(shroud, hit-angle)

… change to ML!

- The first problem we faced is single Ar39 decays:
  - Single decays constitute the **98%** of **Ar39 background** (Poisson)
  - The trigger rate of **2.4kHz** is mainly because of single Ar39,
- We should first define a strategy to reject them, and then move to more sophisticated classification.

- **At the beginning**, we thought we can simply perform a **cut on NPE** to reject all of them. Simulation of Single Ar39 has NPE in [0, 60].
- Also ~30% of simulated muons deposit <=60PE.

- **Requirements**:
  1) Reject single Ar39 decays as much as possible,
  2) Save very low-energy muons (PE<=60), if possible.

Features

- Instance **format** by using 2 Spatial Maps:
  InnerSlice0, ..., InnerSlice11, OuterSlice0, ..., OuterSlice19
- 32 int values, each is the nr of photons that are detected in that slice

- Features for each shroud:
  - <u>PEDetected</u>: sum PE detected in the shroud
  - <u>NactiveSlices</u>: number of slices of the shroud wt >=1PE
  - <u>MeanNPE</u>: PEDetected / NSlices
  - <u>StdNPE</u>: std({PE | for PE>0 in shroud})
  - <u>SpatialRange</u>: difference max - min active slice
  - <u>SpatialVar</u>: var({ID**PE | ID of slice wt PE})
  - <u>SpatialStd</u>: std({ID**PE | ID of slice wt PE})
- 7 inner features, 7 outer features (tot. 14 features)

Feature Selection (1)

- Correlation Matrix

	PEDetected_inner	NActiveSlices_inner	MeanNPE_inner	StdNPE_inner	SpatialRange_inner	SpatialVar_inner	SpatialStd_inner	PEDetected_outer	NActiveSlices_outer	MeanNPE_outer	StdNPE_outer	SpatialRange_outer	SpatialVar_outer	SpatialStd_outer	y
PEDetected_inner	1	0.77	1	0.98	0.63	0.31	0.57	0.047	-0.0021	0.066	0.06	-9.2e-06	0.011	-0.017	-0.19
NActiveSlices_inner	0.77	1	0.77	0.78	0.89	0.5	0.86	0.065	0.0086	0.088	0.069	-0.0081	0.0036	-0.038	-0.27
MeanNPE_inner	1	0.77	1	0.98	0.63	0.31	0.57	0.047	-0.0021	0.066	0.06	-9.2e-06	0.011	-0.017	-0.19
StdNPE_inner	0.98	0.78	0.98	1	0.65	0.31	0.6	0.036	-0.023	0.056	0.047	-0.021	-0.00063	-0.039	-0.23
SpatialRange_inner	0.63	0.89	0.63	0.65	1	0.81	0.99	0.074	0.028	0.094	0.077	0.025	0.03	-0.0041	-0.23
SpatialVar_inner	0.31	0.5	0.31	0.31	0.81	1	0.83	0.081	0.063	0.092	0.083	0.068	0.059	0.05	-0.079
SpatialStd_inner	0.57	0.86	0.57	0.6	0.99	0.83	1	0.061	0.013	0.08	0.061	0.01	0.024	-0.017	-0.24
PEDetected_outer	0.047	0.065	0.047	0.036	0.074	0.081	0.061	1	0.83	1	0.98	0.57	0.28	0.42	0.36
NActiveSlices_outer	-0.0021	0.0086	-0.0021	-0.023	0.028	0.063	0.013	0.83	1	0.83	0.79	0.72	0.36	0.59	0.43
MeanNPE_outer	0.066	0.088	0.066	0.056	0.094	0.092	0.08	1	0.83	1	0.98	0.57	0.27	0.42	0.34
StdNPE_outer	0.06	0.069	0.06	0.047	0.077	0.083	0.061	0.98	0.79	0.98	1	0.55	0.26	0.41	0.34
SpatialRange_outer	-9.2e-06	-0.0081	-9.2e-06	-0.021	0.025	0.068	0.01	0.57	0.72	0.57	0.55	1	0.84	0.95	0.41
SpatialVar_outer	0.011	0.0036	0.011	-0.00063	0.03	0.059	0.024	0.28	0.36	0.27	0.26	0.84	1	0.94	0.27
SpatialStd_outer	-0.017	-0.038	-0.017	-0.039	-0.0041	0.05	-0.017	0.42	0.59	0.42	0.41	0.95	0.94	1	0.36
y	-0.19	-0.27	-0.19	-0.23	-0.23	-0.079	-0.24	0.36	0.43	0.34	0.34	0.41	0.27	0.36	1

Feature Selection (2)

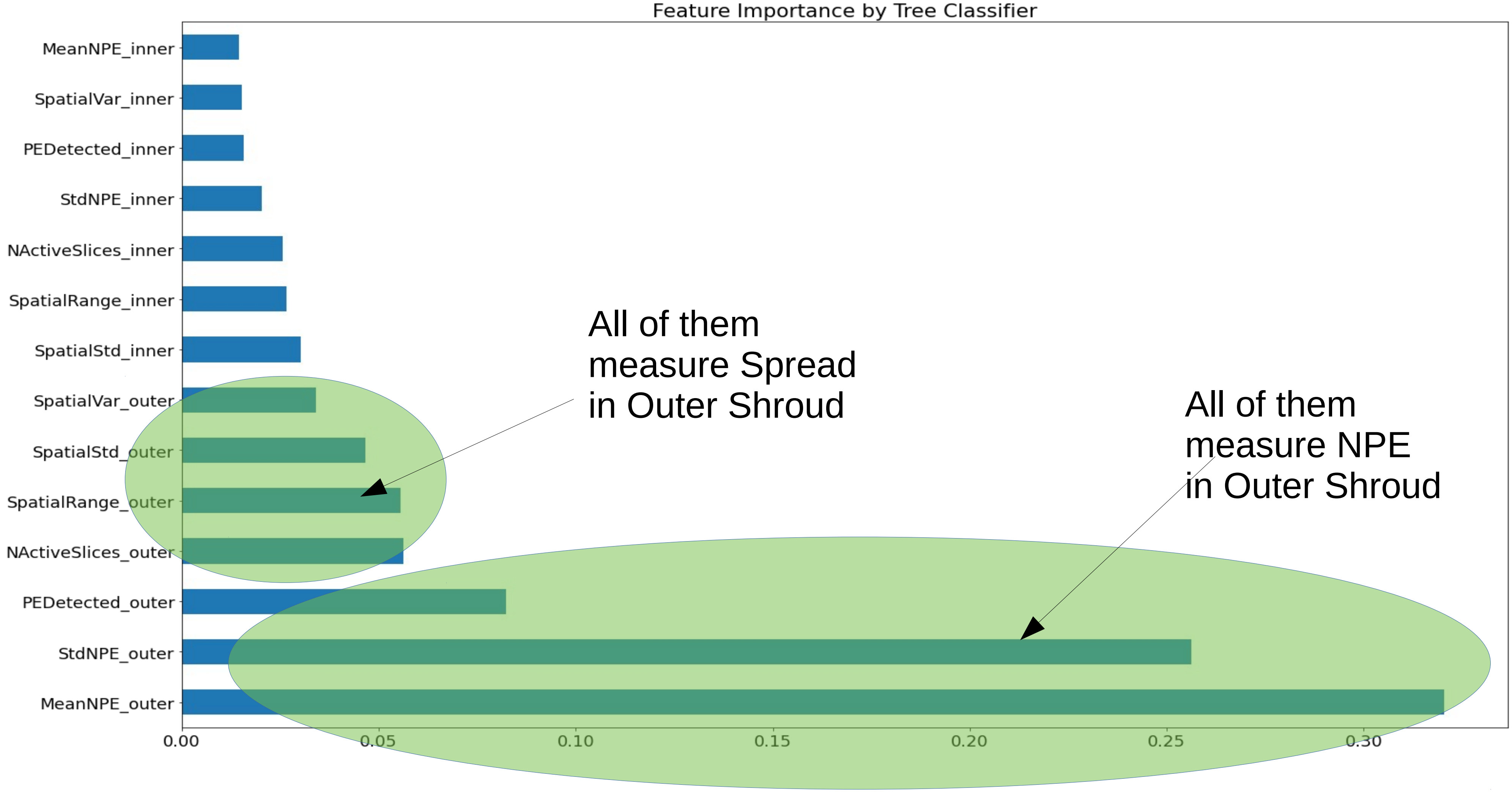
- F-Score: “correlation” with label y

	Specs	Score
8	NActiveSlices_outer	1916.874748
11	SpatialRange_outer	1649.754009
7	PEDetected_outer	1263.550430
13	SpatialStd_outer	1239.324026
10	StdNPE_outer	1070.927875
9	MeanNPE_outer	1050.389635
1	NActiveSlices_inner	675.351404
12	SpatialVar_outer	640.842699
6	SpatialStd_inner	526.291369
3	StdNPE_inner	458.870149
4	SpatialRange_inner	453.021880
0	PEDetected_inner	311.159887
2	MeanNPE_inner	311.159887
5	SpatialVar_inner	51.663484

- **Observation**: the first features are all “outer” features.

Feature Selection (3)

- Train a Dtree and see what features are selected first



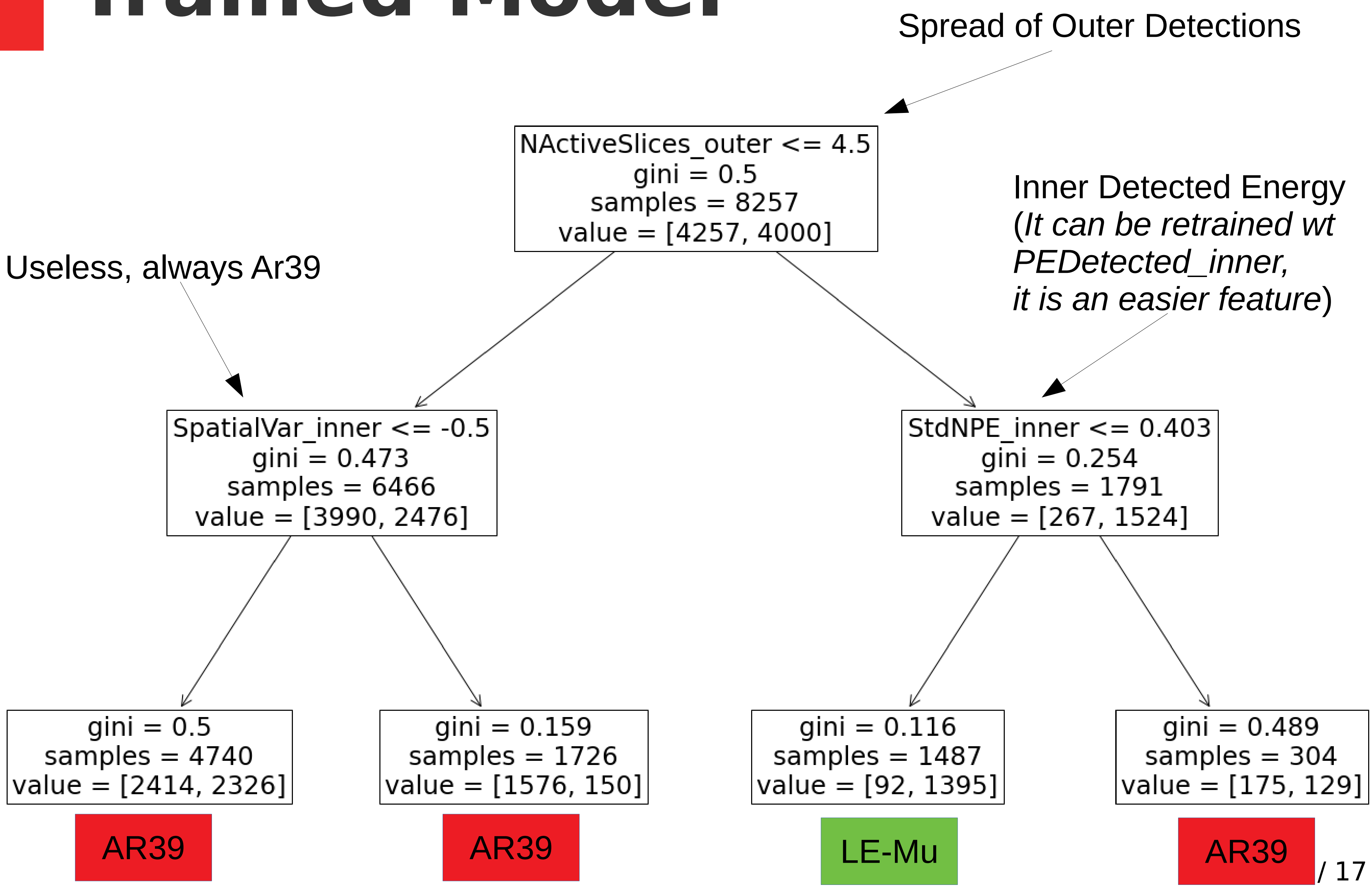
<u>First attempt</u>: remove all features wt correlation>.95. **8 Selected Features:** NActiveSlices_inner, StdNPE_inner, SpatialVar_inner, SpatialStd_inner, NActiveSlices_outer, StdNPE_outer, SpatialVar_outer, SpatialStd_outer

DTree – Training

- 8 Features:
  - NActiveSlices_inner, StdNPE_inner, SpatialVar_inner, SpatialStd_inner,
  - NActiveSlices_outer, StdNPE_outer, SpatialVar_outer, SpatialStd_outer
- Training data: 4000 LE Muons, 4257 Ar39
- Hyperparameters:
  - Criterion
  - Max Depth
  - Min Sample Leaf

- Aim: “Reject as much 1Ar39 as possible, saving muons if possible”
- Score for training: “purity”, it takes care of TP and FP!

Trained Model



DTree – Test on unbalanced data

- Test Set: **626** LE Muons, **2M** Ar39 (*unseen data*)

- Evaluation of **LE Classification** (<=60PE):
  - TPR (efficiency) = 38.5% (241 / 626)
  - FPR = 3.48% (69601 / 2M)

- **Consideration FPR:**
- This is a first-level classification (we are not considering Pileups), we will extend it with other classifiers!
- Assuming this FPR, the Ar39 Background:
- 0.9883(1Ar39) + .0117(+Ar39) would be
  < 0.9883*0.0348 + 0.0117 = 4.61%
- Original Trigger Rate: 2353 Hz, Reduced Trigger Rate: 109 Hz

The big picture