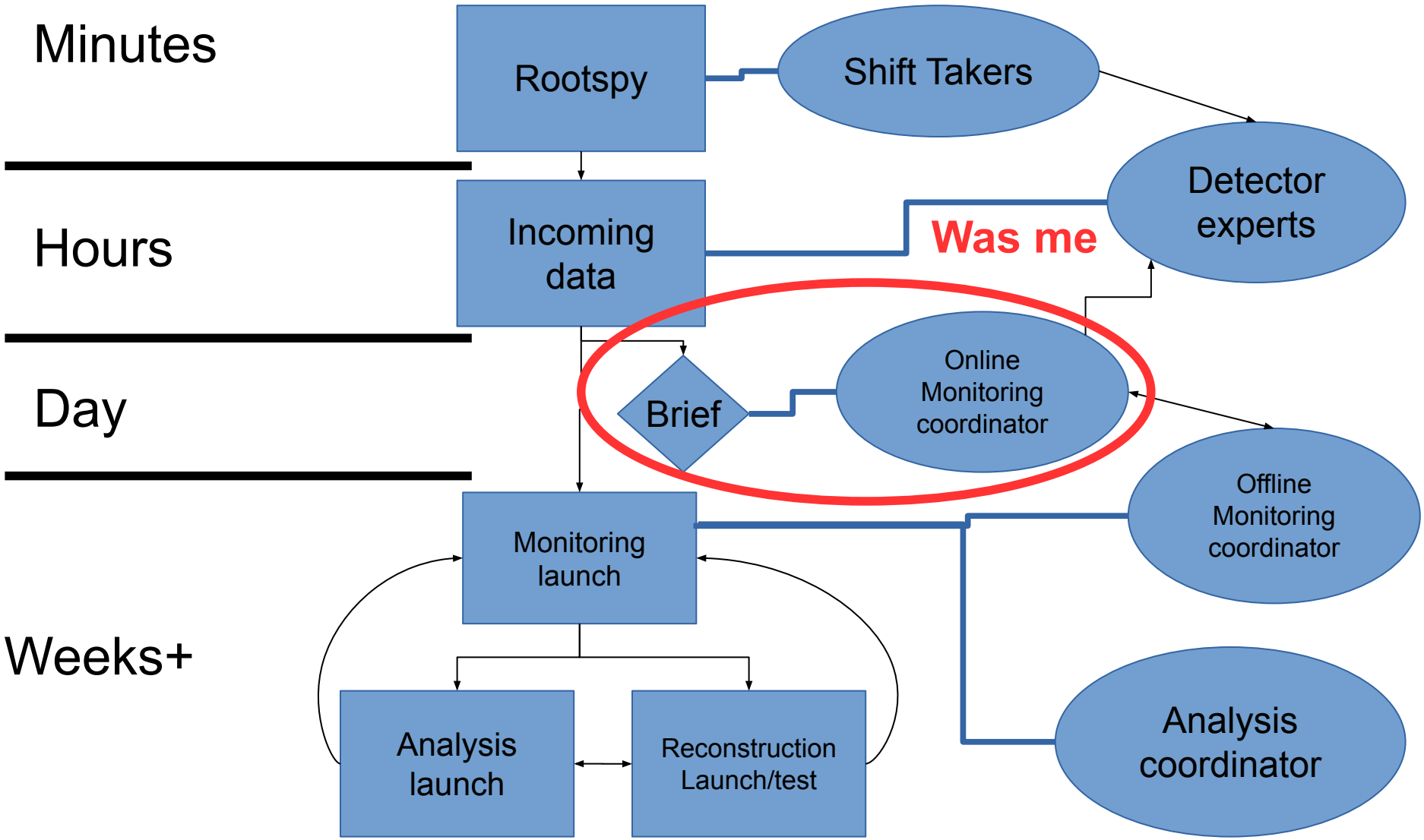
AI: Data Quality Monitoring

Thomas Britton

From DAQ to ANA



The Challenge

- **Every run** produces an initial **22 plots**. More thorough monitoring is performed offline and produces **109 plots**. With a run lasting **~3 hours** every day there are **between ~175 and 875** plots to look at.
  - To preserve sanity I looked at closer to 175 plots, but there is no reason a machine couldn't aid in looking at all of them…
- Often times a single plot being "off" is not an indication of problems. Need to look at all the plots to determine cause and severity
  - Trigger studies: Often look like big problems but are not. Can be hard to catch when shift logs have scant details

Let's Play

Ladies and gentlemen: the story you are about to hear is true. Only the names have been changed to protect the innocent.

*Logbook searches not included

Questions to keep in mind

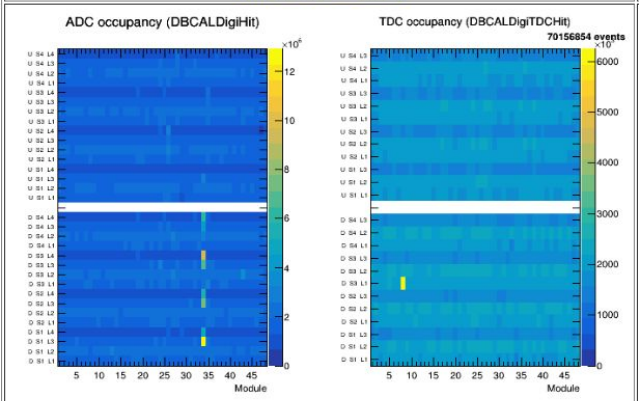
What is the problem? Is it isolated? Is this due to running conditions or a test?
Should we stop taking data/call the shift crew?

It's 5:15am the email has just come in...

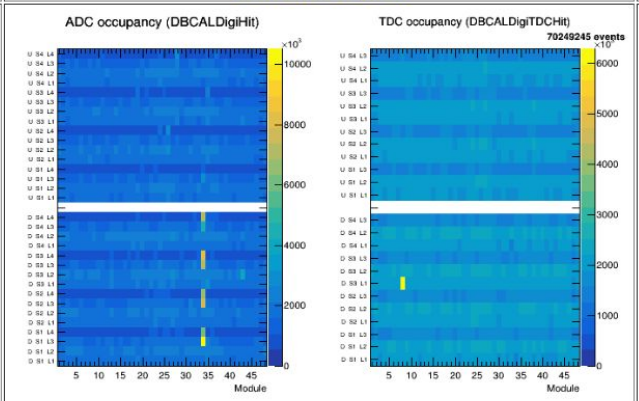
Let's Play

Anomaly detected! BCAL occupancy changed

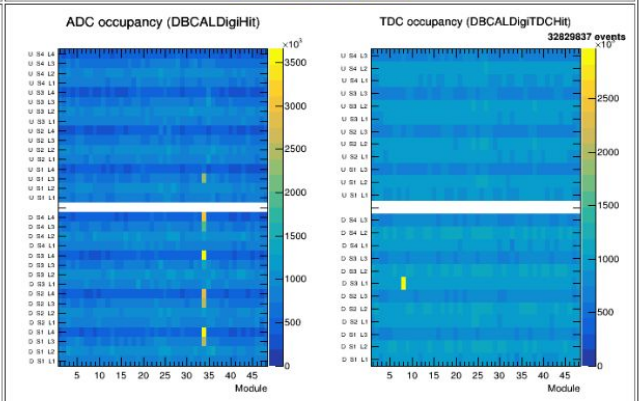
Run 042382 info



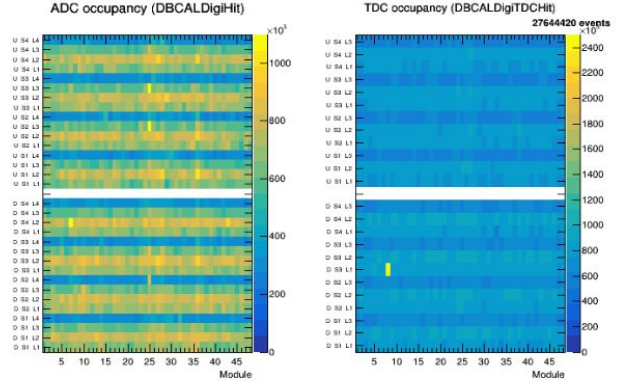
Run 042381 info



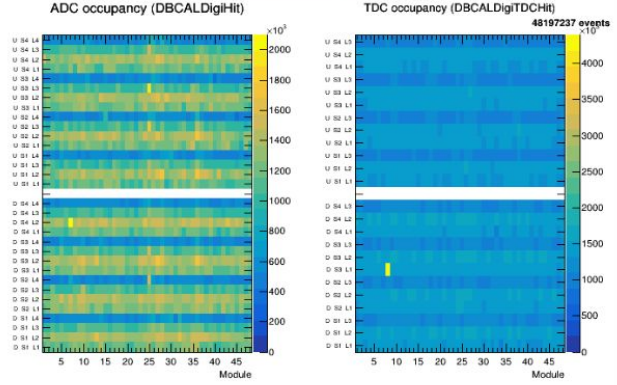
Run 042374 info



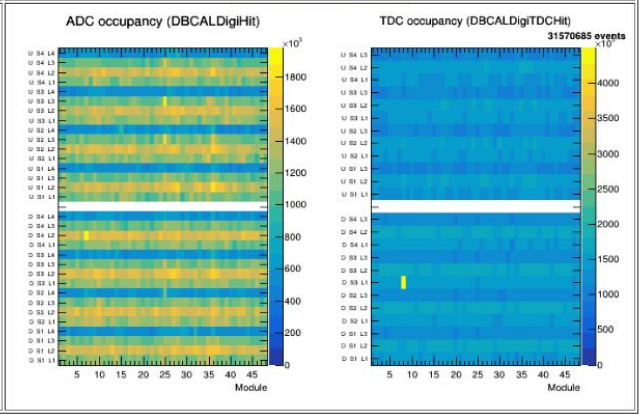
Run 042352 info



Run 042351 info



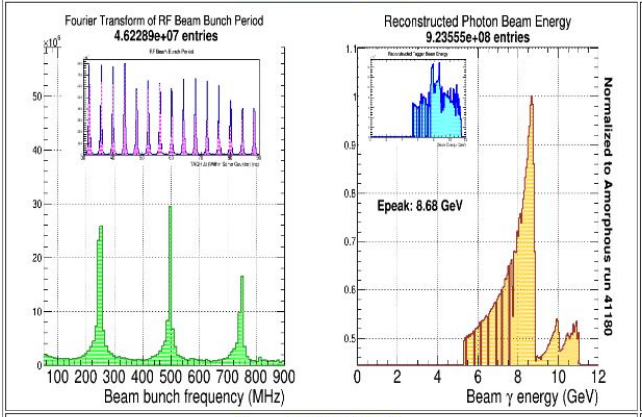
Run 042350 info



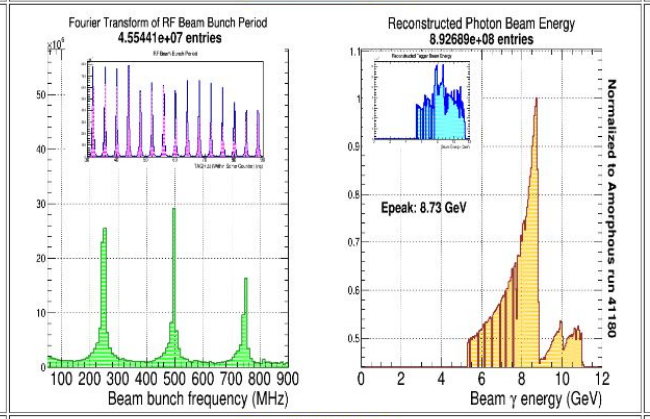
Let's Play

Nothing looks really off with the beam

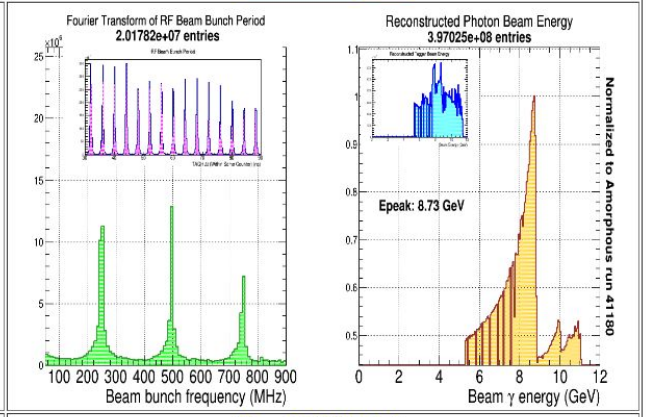
Run 042382 info



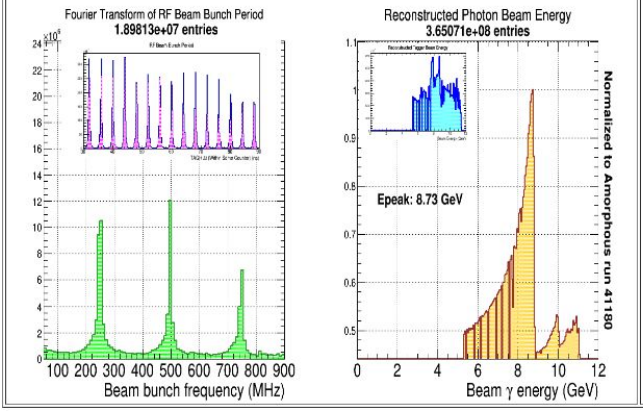
Run 042381 info



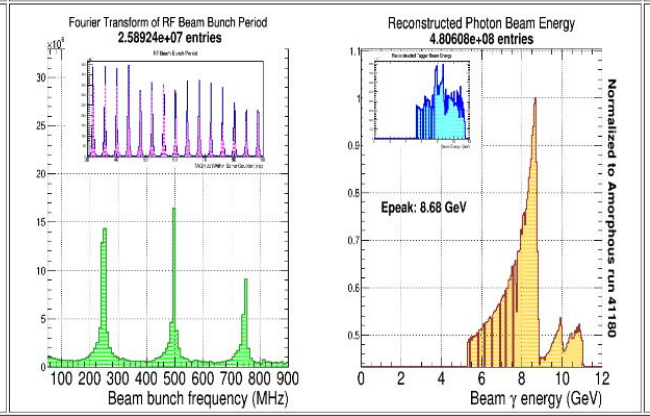
Run 042374 info



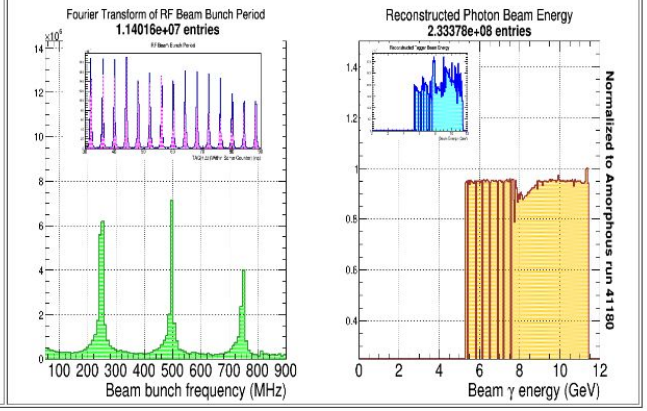
Run 042352 info



Run 042351 info

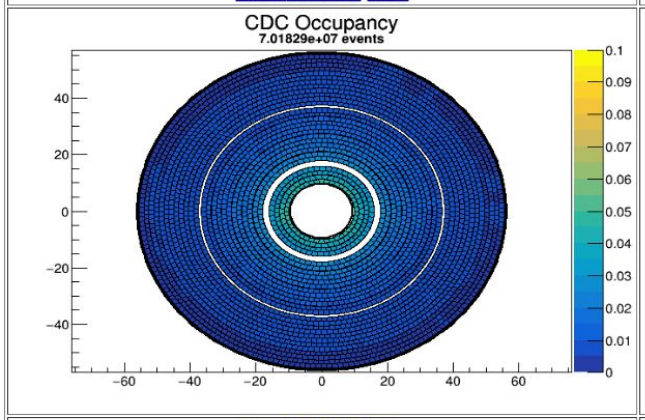
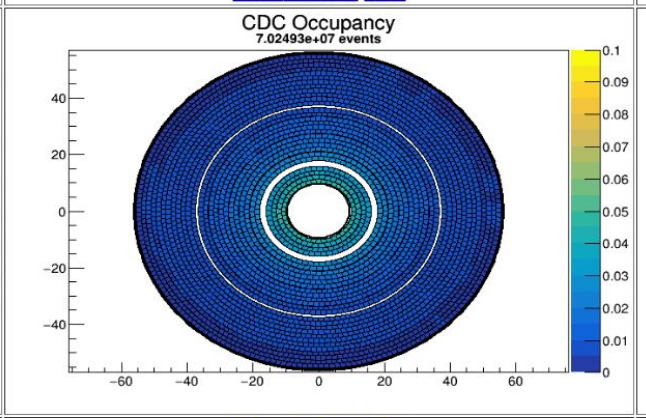
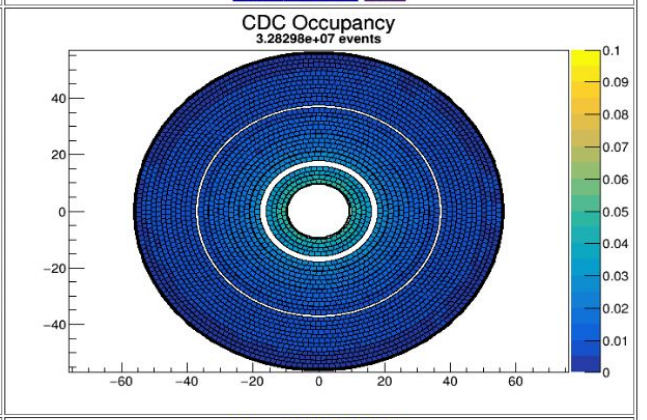
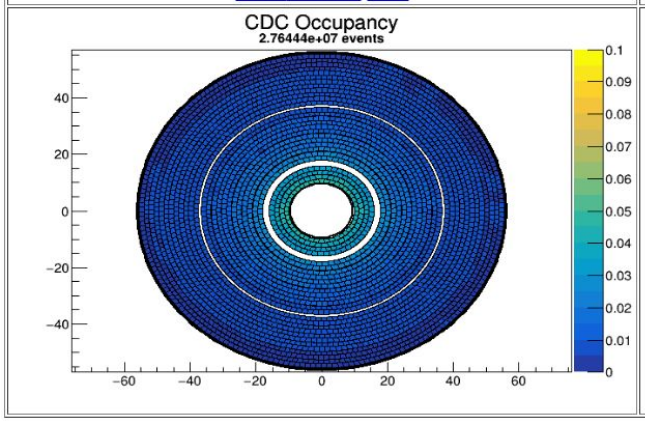
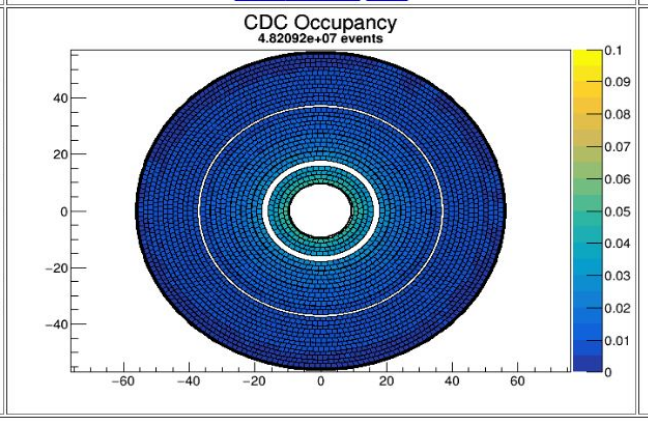
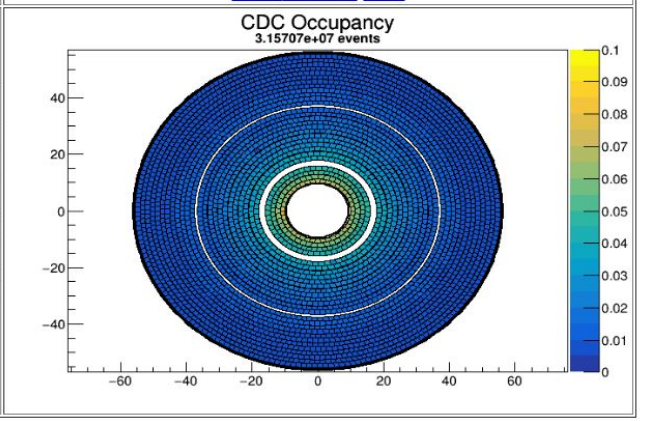


Run 042350 info



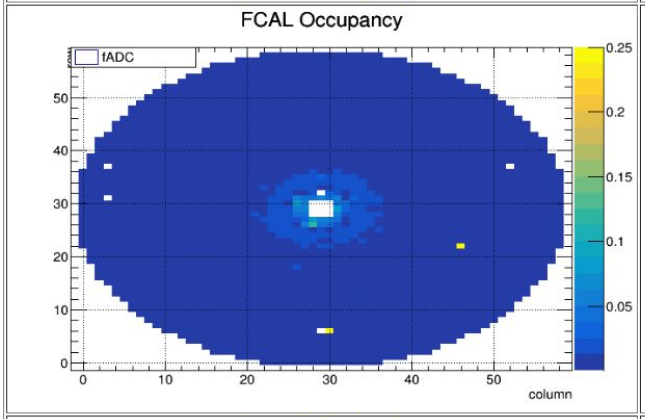
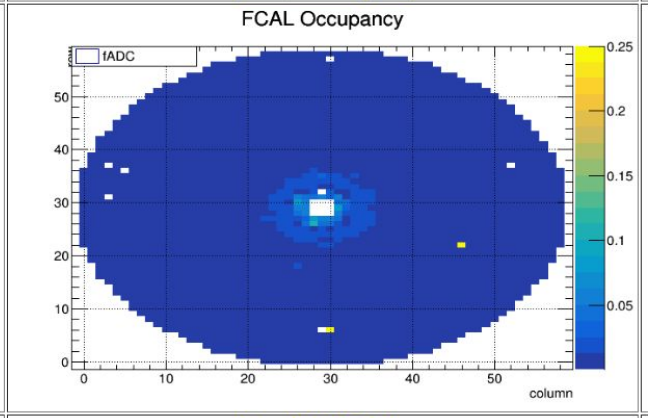
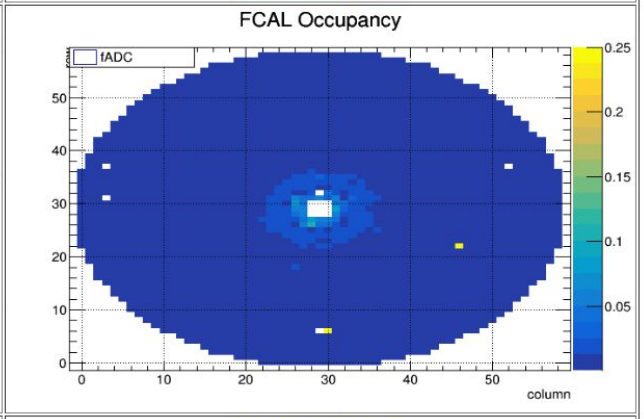
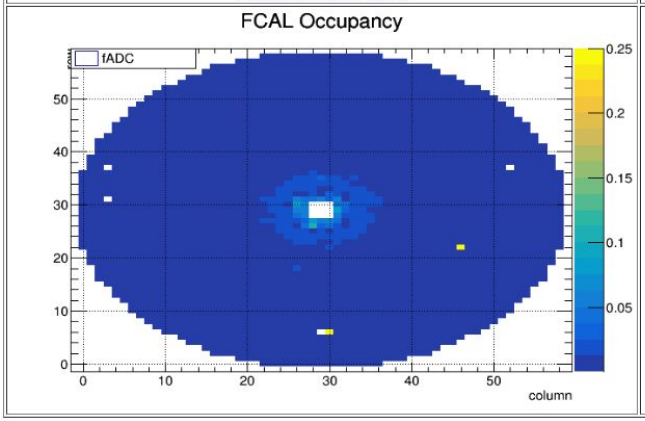
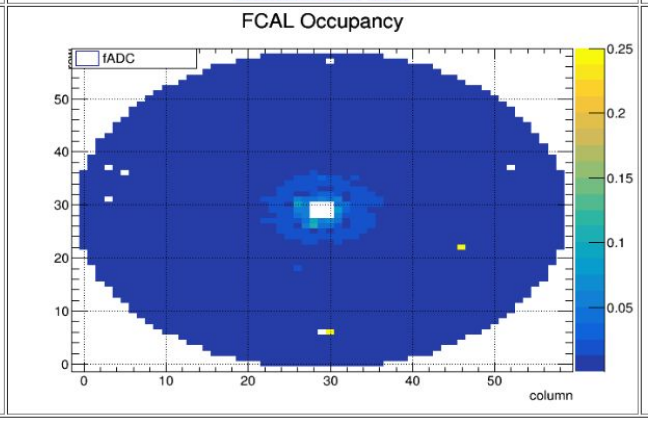
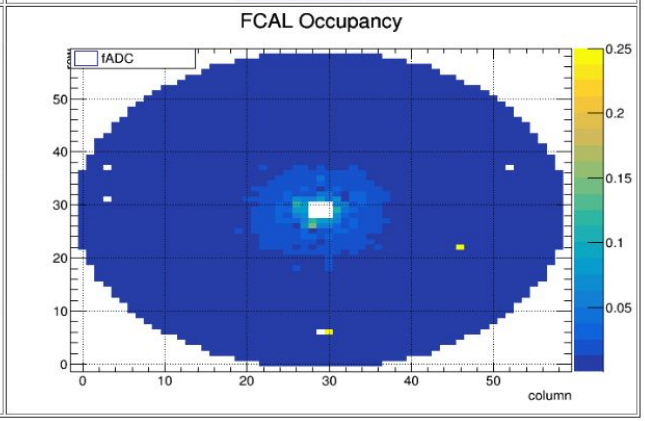
Let's Play

CDC "dims"….hmmmm. Did you remember 42351 from the BCAL looked normal? Probably uncorrelated…..

Run 042382 info	Run 042381 info	Run 042374 info
		
Run 042352 info	**Run 042351 info**	**Run 042350 info**
		

Let's Play

FCAL looks normal. Note a few new dead modules….

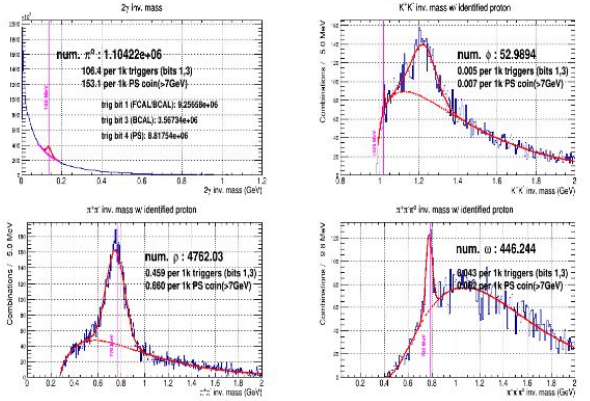
Run 042382 info	Run 042381 info	Run 042374 info
		
Run 042352 info	**Run 042351 info**	**Run 042350 info**
		

Let's Play

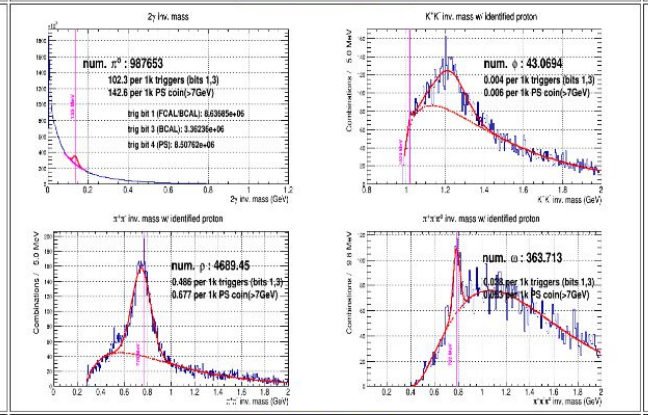
13 plots later….

Physics signals still in tolerance

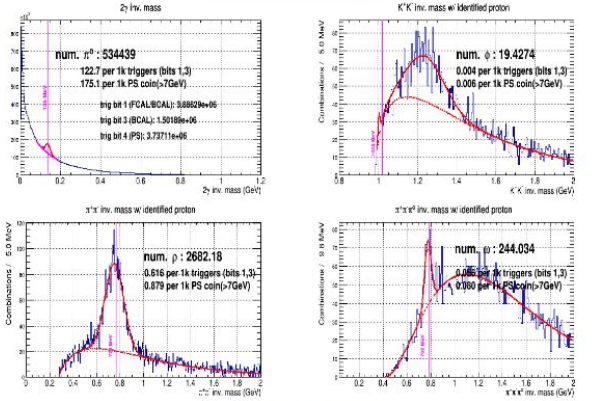
Run 042382 info



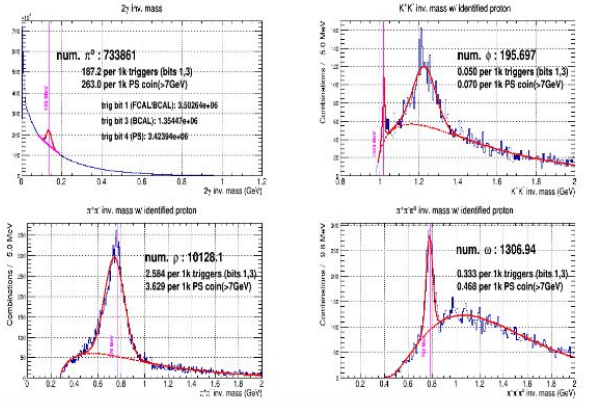
Run 042381 info



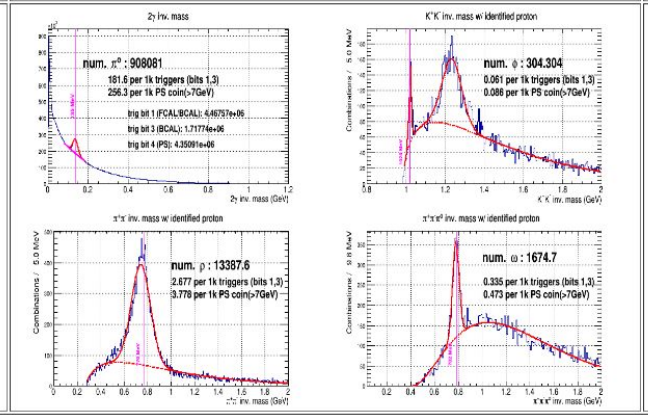
Run 042374 info



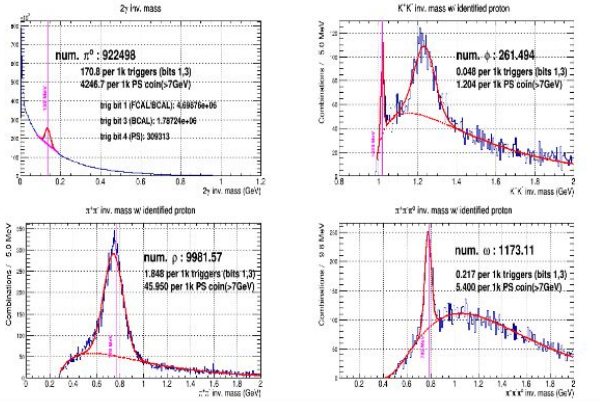
Run 042352 info



Run 042351 info



Run 042350 info

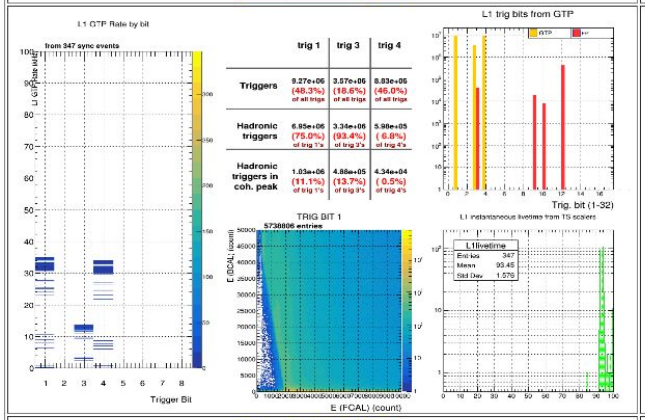


Let's Play

5 plots later...

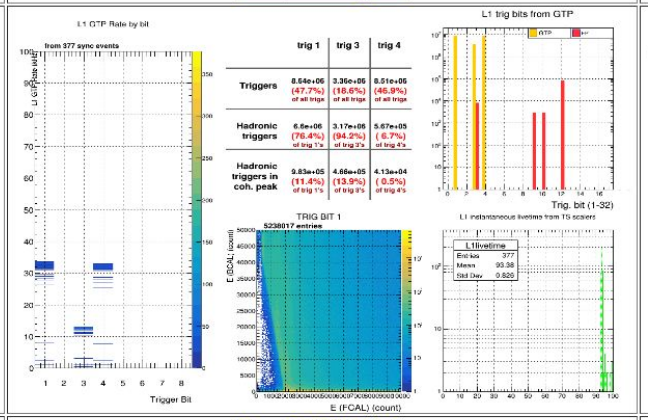
Increased dead time at 42351

Run 042382 info



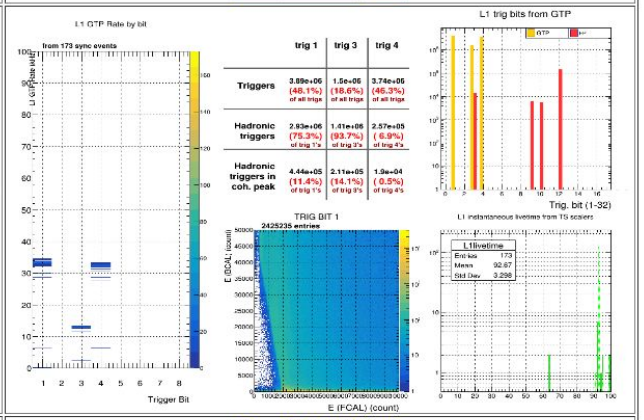
	trig 1	trig 3	trig 4
Triggers	9.27e+06 (46.3%) of all trigs	3.67e+06 (18.6%) of all trigs	8.83e+06 (46.0%) of all trigs
Hadronic triggers	6.95e+06 (75.0%) of trig 1's	3.34e+06 (93.4%) of trig 3's	5.98e+05 (6.8%) of trig 4's
Hadronic triggers in coh. peak	1.03e+06 (11.1%) of trig 1's	4.88e+05 (13.7%) of trig 3's	4.34e+04 (0.5%) of trig 4's

Run 042381 info



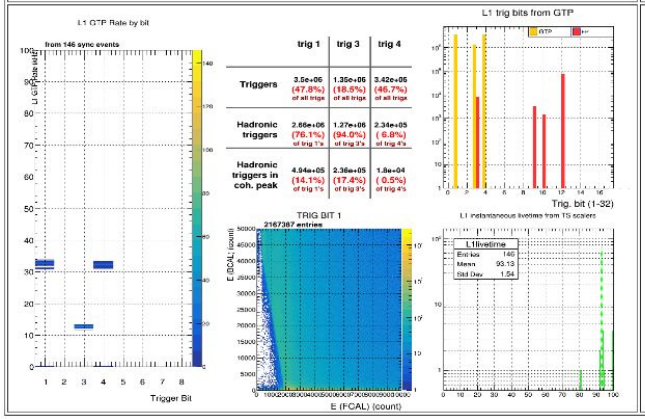
	trig 1	trig 3	trig 4
Triggers	8.54e+06 (47.7%) of all trigs	3.36e+06 (18.6%) of all trigs	8.51e+06 (46.9%) of all trigs
Hadronic triggers	6.5e+06 (76.4%) of trig 1's	3.17e+06 (94.2%) of trig 3's	5.67e+05 (6.7%) of trig 4's
Hadronic triggers in coh. peak	9.83e+05 (11.4%) of trig 1's	4.66e+05 (13.9%) of trig 3's	4.13e+04 (0.5%) of trig 4's

Run 042374 info



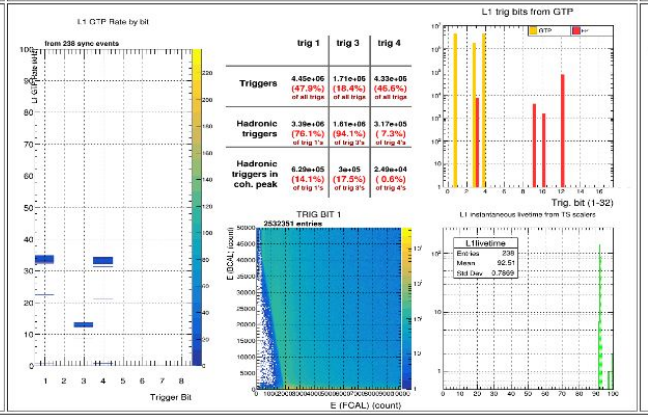
	trig 1	trig 3	trig 4
Triggers	3.89e+06 (48.1%) of all trigs	1.5e+06 (18.6%) of all trigs	3.74e+06 (46.3%) of all trigs
Hadronic triggers	2.93e+06 (75.3%) of trig 1's	1.41e+06 (93.7%) of trig 3's	2.57e+05 (6.9%) of trig 4's
Hadronic triggers in coh. peak	4.44e+05 (11.4%) of trig 1's	2.11e+05 (14.1%) of trig 3's	1.9e+04 (0.5%) of trig 4's

Run 042352 info



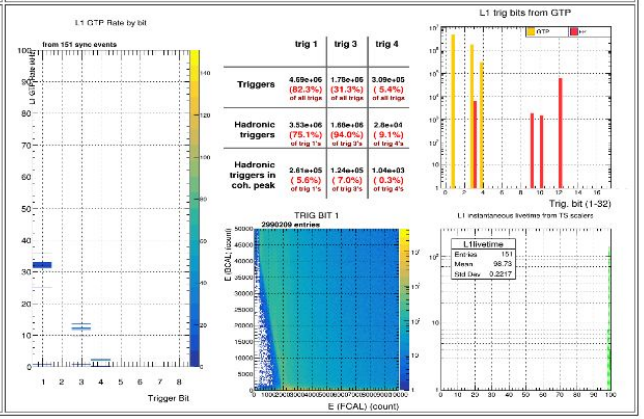
	trig 1	trig 3	trig 4
Triggers	3.6e+06 (47.8%) of all trigs	1.39e+06 (18.5%) of all trigs	3.42e+06 (46.7%) of all trigs
Hadronic triggers	2.68e+06 (76.1%) of trig 1's	1.27e+06 (94.0%) of trig 3's	2.34e+05 (6.8%) of trig 4's
Hadronic triggers in coh. peak	4.94e+05 (14.1%) of trig 1's	2.36e+05 (17.4%) of trig 3's	1.8e+04 (0.5%) of trig 4's

Run 042351 info



	trig 1	trig 3	trig 4
Triggers	4.45e+06 (47.9%) of all trigs	1.71e+06 (18.4%) of all trigs	4.33e+06 (46.6%) of all trigs
Hadronic triggers	3.39e+06 (76.1%) of trig 1's	1.61e+06 (94.1%) of trig 3's	3.17e+05 (7.3%) of trig 4's
Hadronic triggers in coh. peak	6.29e+05 (14.1%) of trig 1's	3e+05 (17.5%) of trig 3's	2.49e+04 (0.6%) of trig 4's

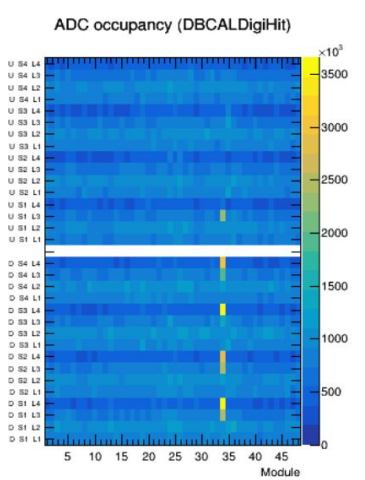
Run 042350 info

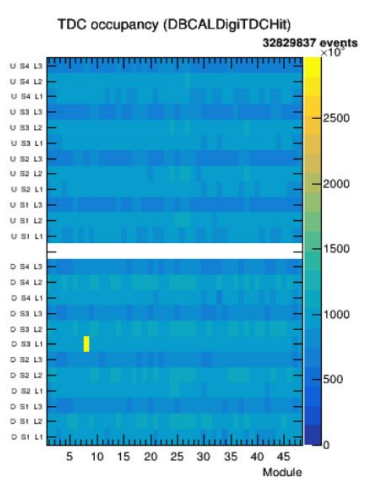


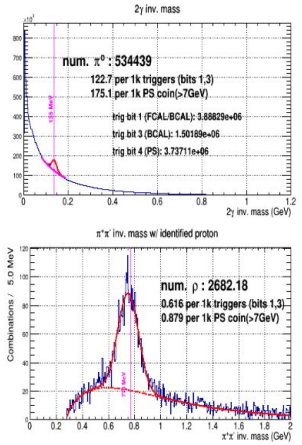
	trig 1	trig 3	trig 4
Triggers	4.69e+06 (82.3%) of all trigs	1.79e+06 (31.3%) of all trigs	3.09e+05 (5.4%) of all trigs
Hadronic triggers	3.53e+06 (75.1%) of trig 1's	1.68e+06 (94.0%) of trig 3's	2.8e+04 (9.1%) of trig 4's
Hadronic triggers in coh. peak	2.61e+05 (5.6%) of trig 1's	1.24e+05 (7.0%) of trig 3's	1.04e+03 (0.3%) of trig 4's

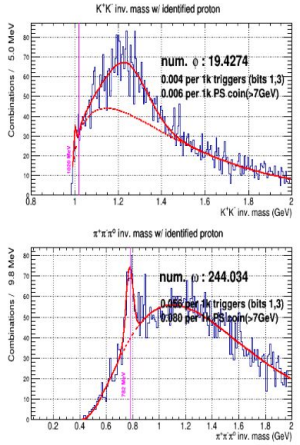
Conclusion

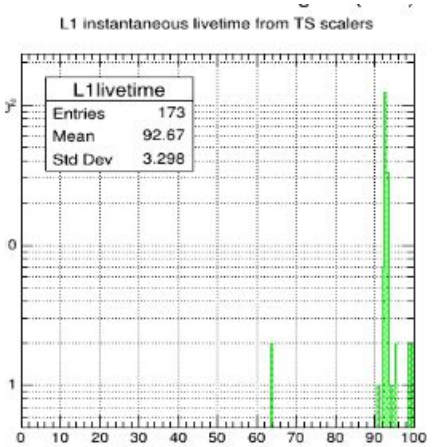
Likely, several BCAL modules have become noisy contributing to an increased deadtime. The reason for occurrence is unknown, but it likely happened during run 42351. It is not significantly impacting physics data. Detector experts should be notified and further investigation should be performed. An opportunistic access to the hall may be needed











Epilogue:

In roughly 24 hours the noisy components were found and replaced. Normal operations resumed

AI

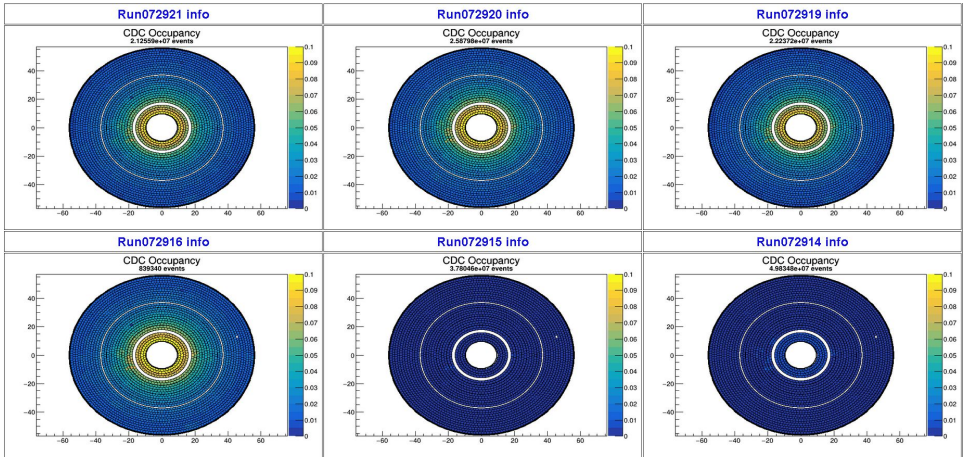
- As demonstrated, what is involved is a lot of pattern recognition and paying attention to the correlations between plots coupled with domain knowledge
  - When plot A looks like X plot B should look like Y but it looks like Z and by looking at plot C probably Q is wrong…
- This is precisely where AI can shine
  - Large, multi-dimensional datasets full of correlations

Considerations

- Many considerations in developing and deploying AI models
  - Data collection:
    - How will you get your training data?
    - How will test data be given?
  - Labeling (in the case of supervised learning):
    - How will you label all of your data?
    - What will that cost?
  - Model development:
    - What requirements does the model need to fulfill? (e.g. inference time, error rates etc)
  - Sociological:
    - How will people interact with the system?
    - Are there any barriers to deployment from the human side?

Data Collection

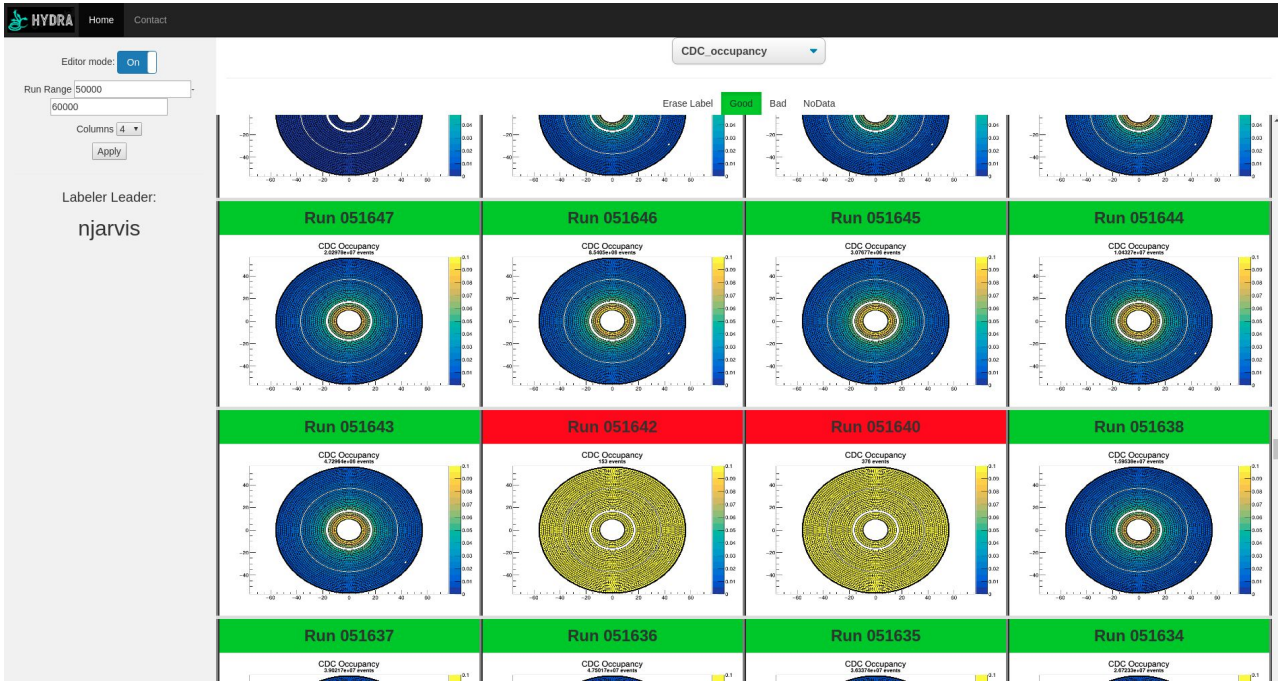
- GlueX already had monitoring plots from all the detectors from all the runs in image format
- Using these is slightly wasteful in terms of data needed for inference
- Get to use all of the developments in machine vision
    - Convolutional Neural Nets
- Cuts down on startup time



Labeling

- Needed an easy way to label vast numbers of images with a low barrier of entry
- Settled on a webpage

Able to label hundreds of plots in minutes

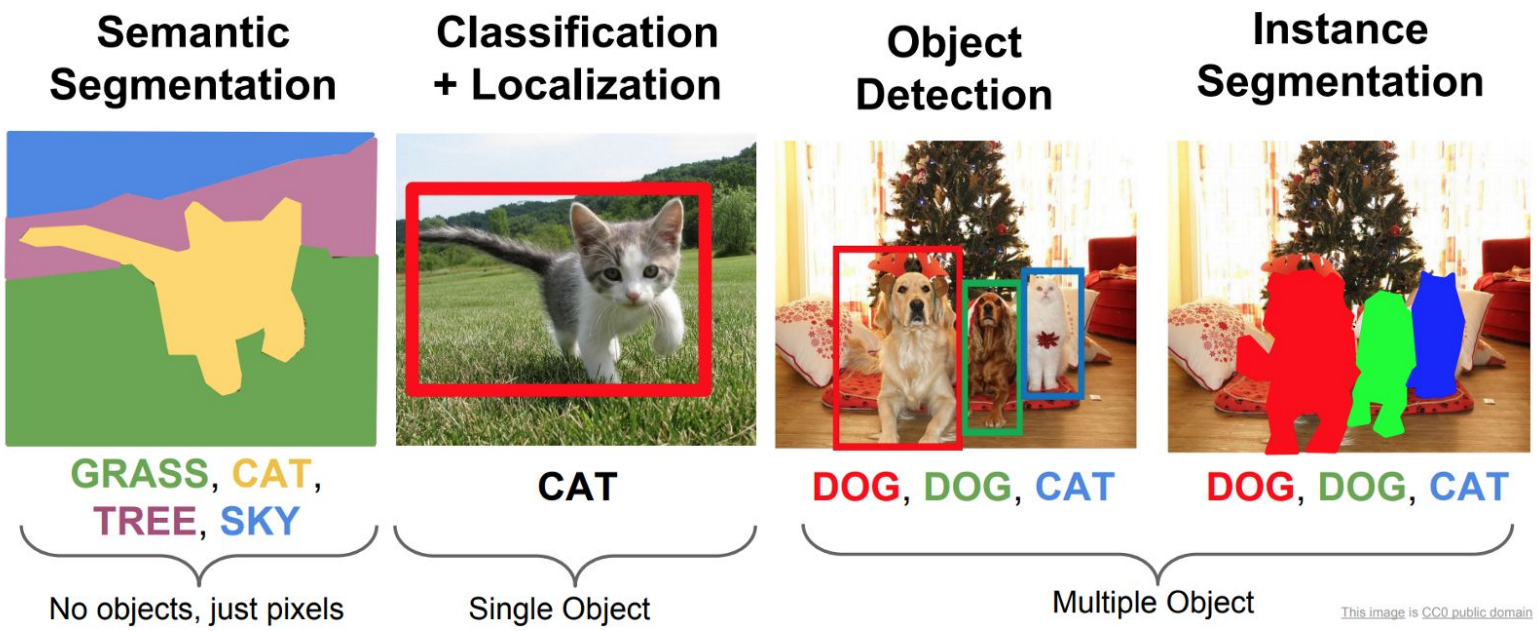


A Challenge

- Labels are horribly unbalanced (Think **~5% bad**).
  - This is ultimately expected as typically detectors are designed to work..
- This can cause problems as the machine may not have enough examples to adequately optimize the weights to classify the lesser labels
  - “It is more advantageous to get really good at identifying cats over dogs or boats as the distribution appears to be heavily skewed to cats”
- Label balancing is the process of repeating examples of lesser labels randomly to make the number of examples equal the biggest label
- This can lead to unnecessarily long training times as it increases, artificially, sample size
  - Strategic undersampling? It doesn’t take many all white images for it to perfectly classify the image as NoData. Maybe curation?
  - Random undersampling of the larger categories to match the smaller categories?
  - Weight specific examples more than others?

Model Development

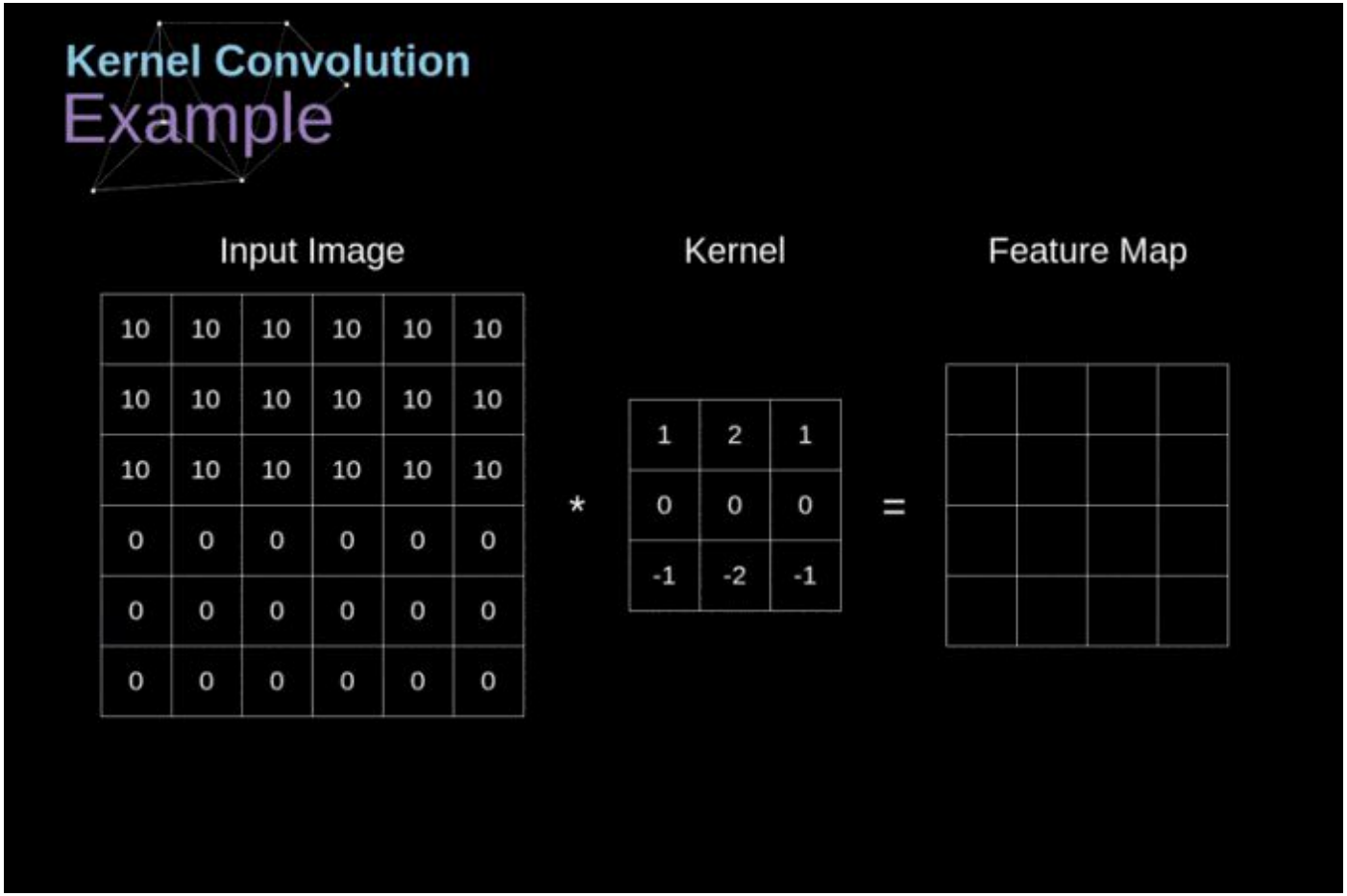
- Because I decided to use the already available images the algorithm choice was pretty clear
  - **Convolutional neural nets**



CNN (math)

https://towardsdatascience.com/

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k]$$



* stride of 1

CNN (kernels)

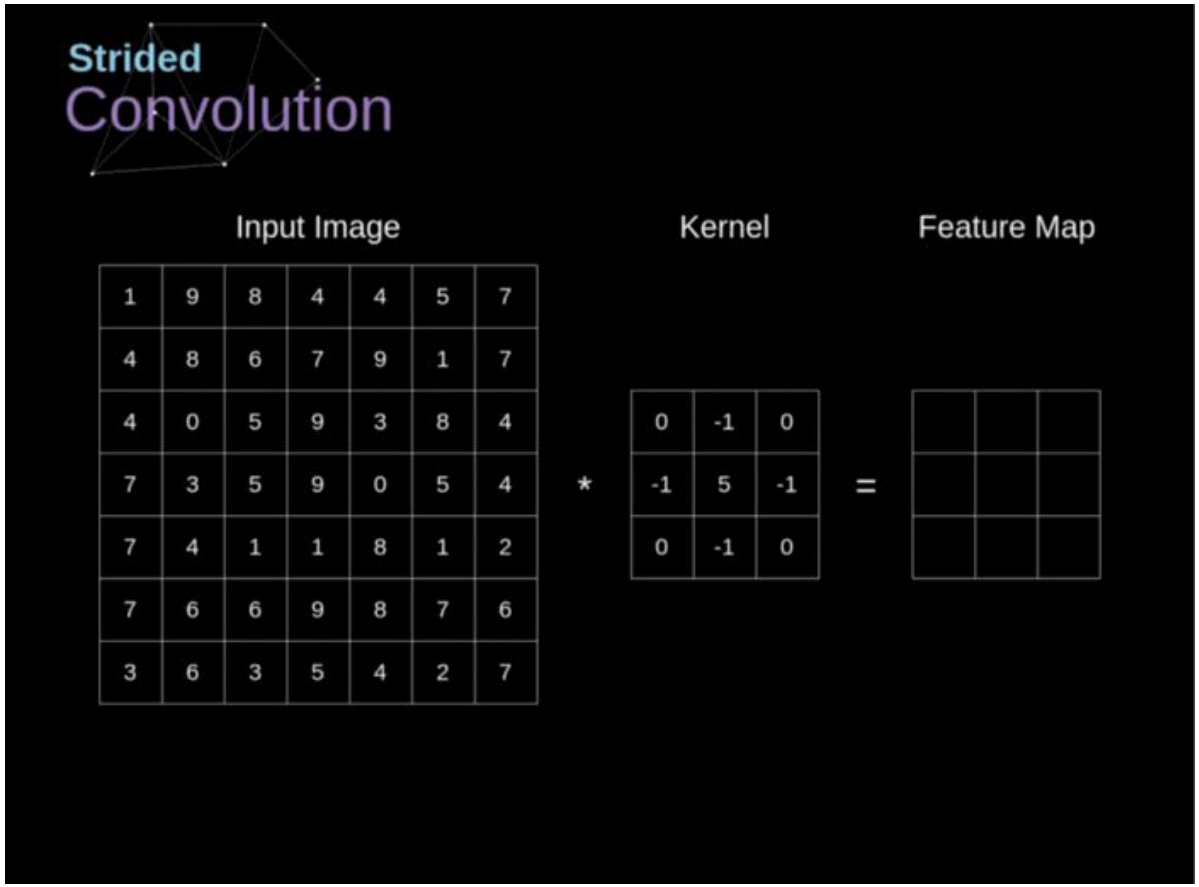
https://towardsdatascience.com/

- Different kernels produce different outputs (features).
  - so you are training a CNN layer to extract different features from the data

CNN (strides)

https://towardsdatascience.com/

- How far to move your matrix window



CNN (pooling)

https://towardsdatascience.com/

- Many types: max, min, avg…

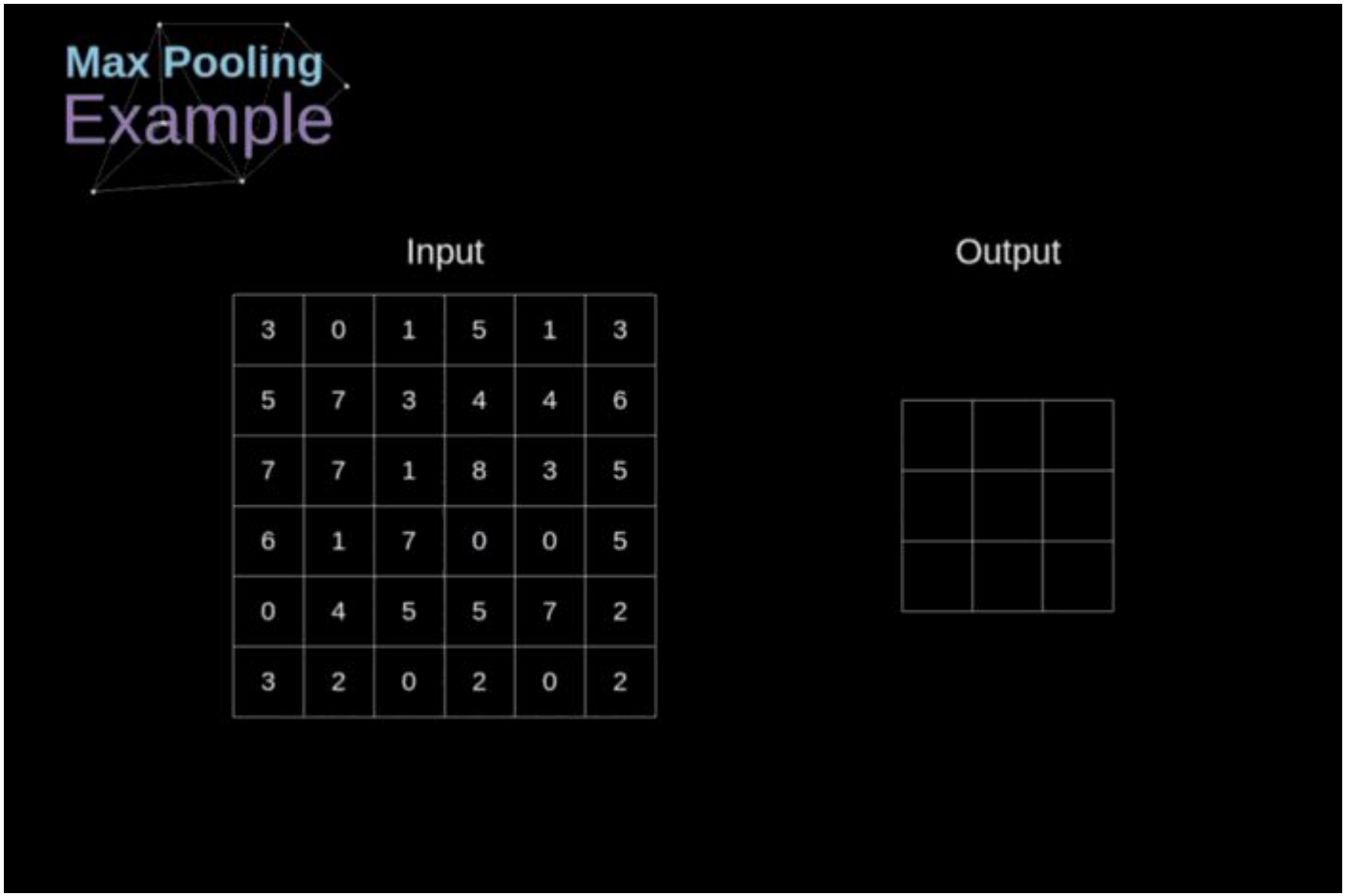
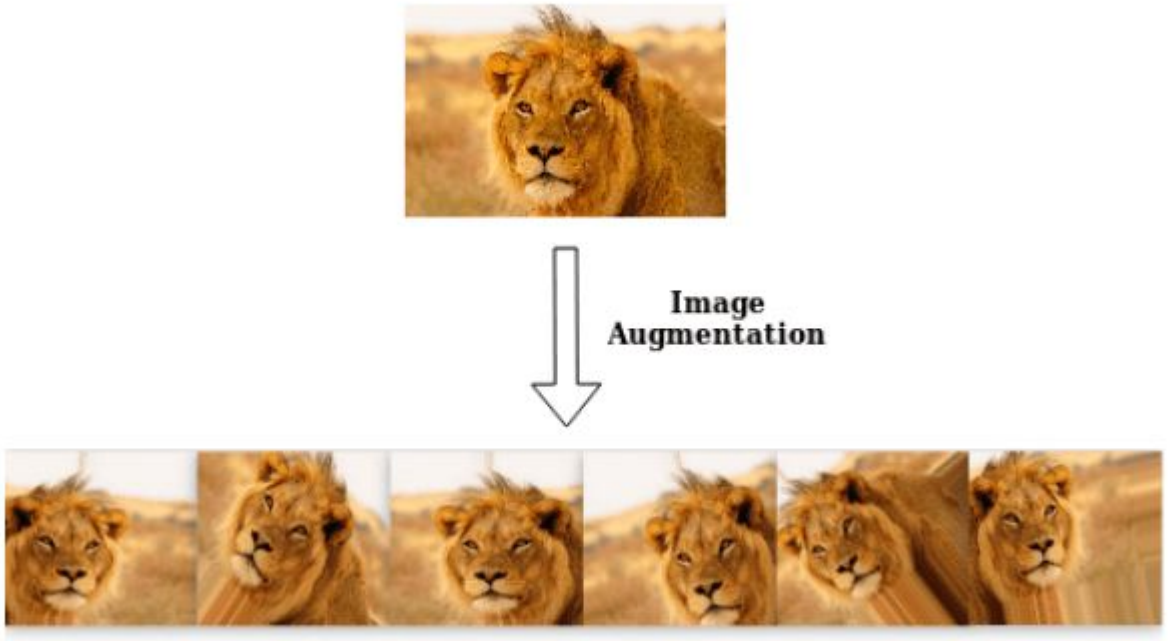


Image Augmentation

- increases dataset and can make a model more accurate
  - can include color shifting etc.

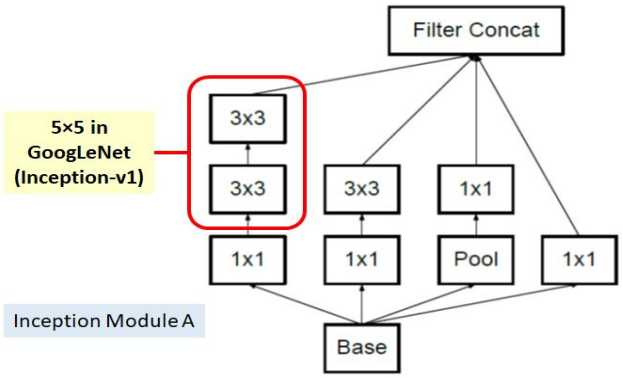


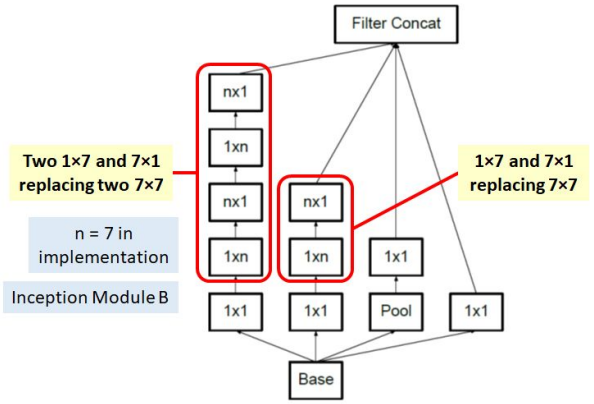
*We don't actually want this for looking at our images as the images are all very very uniform

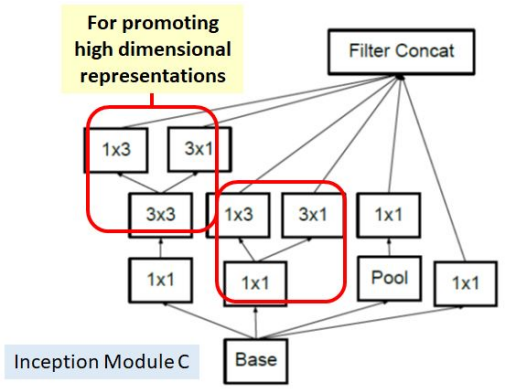
...well maybe color shifting to capture new scales…..

https://towardsdatascience.com/

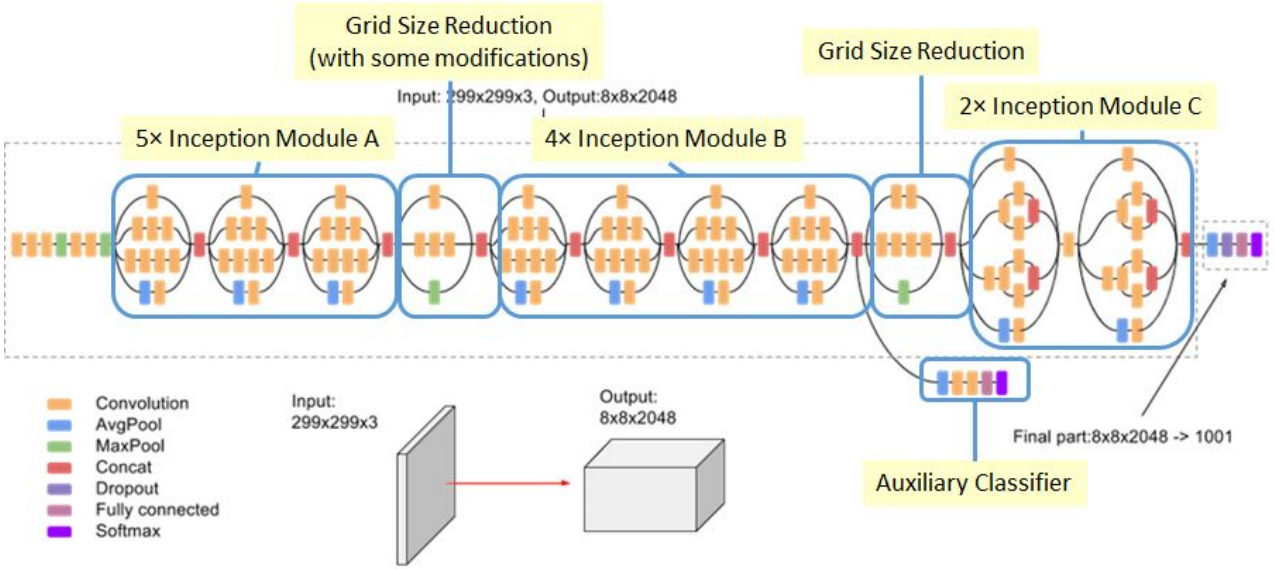
The Inception v3 Network





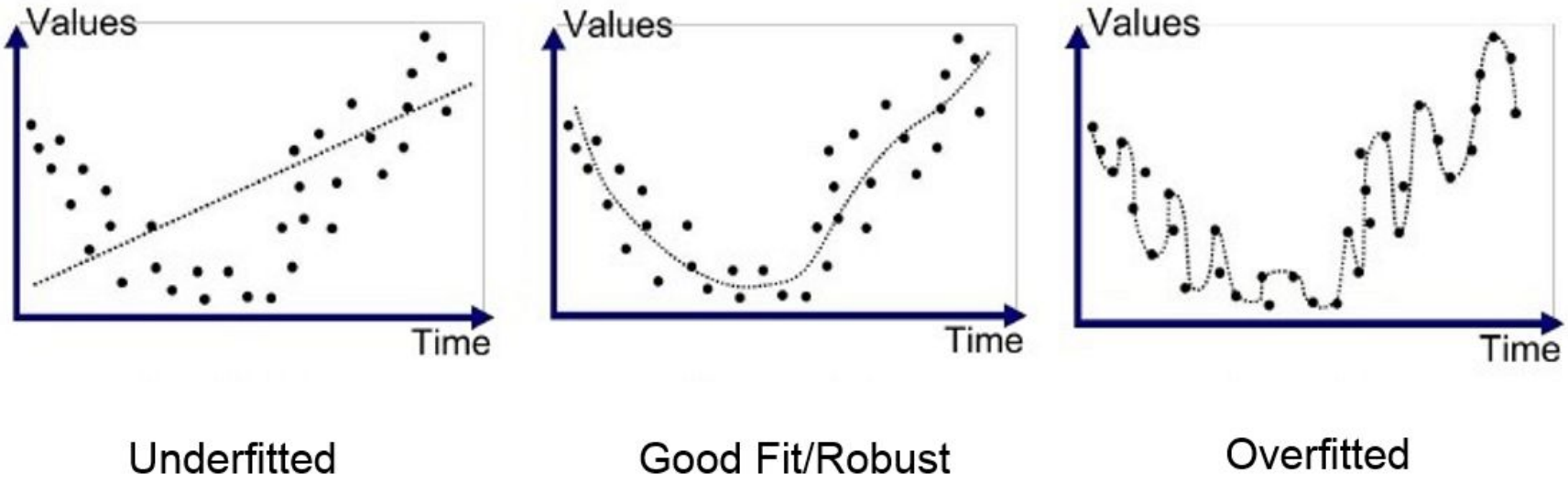


Already in Keras and Tensorflow



Overfitting

- Overfitting a big question



- We have the ability to generate "fake" data
  - May not be feasible to train on

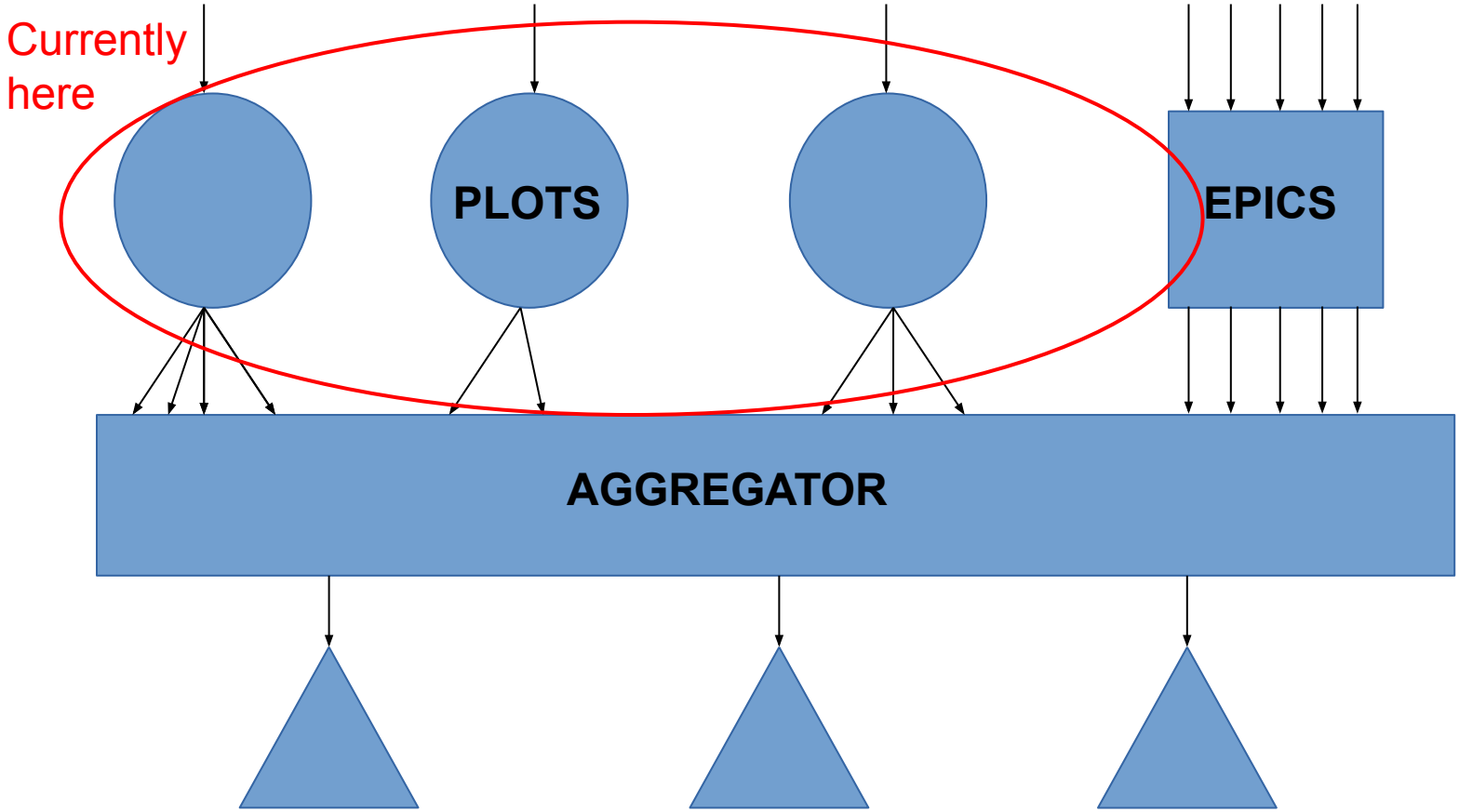
- If the points above were the totality of possibilities…
  - the right most model would be a great, albeit unsatisfying, model

Introducing Hydra

- Hydra aims to be an extensible framework for training and managing A.I. for near real time monitoring
  - If you need it to tell a dog from cat I can have hydra do that, without system modification, now
- Most importantly, Hydra allows me to embrace my inner sloth:

Koboldpress.com

Future Topology

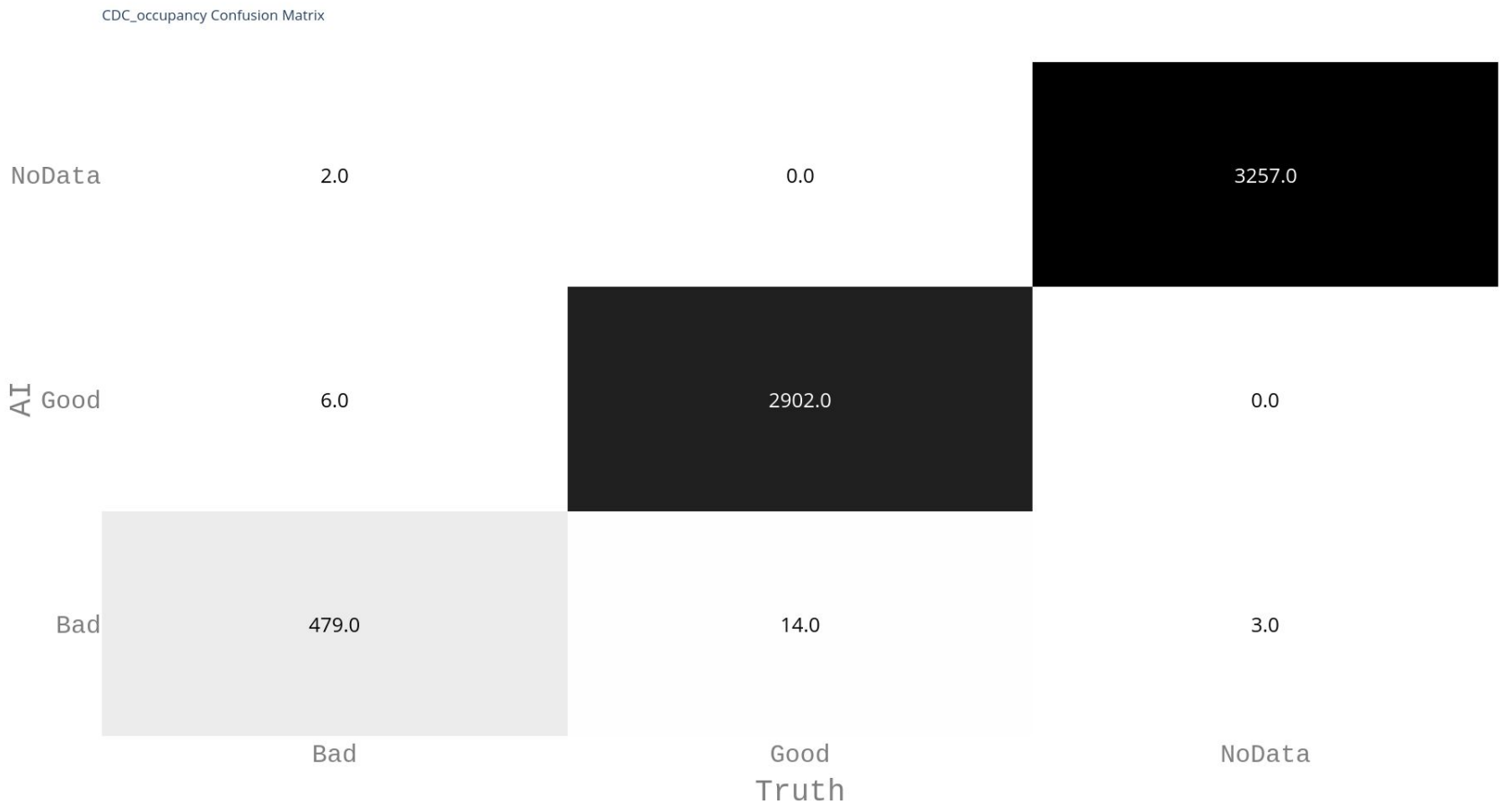


Predictions

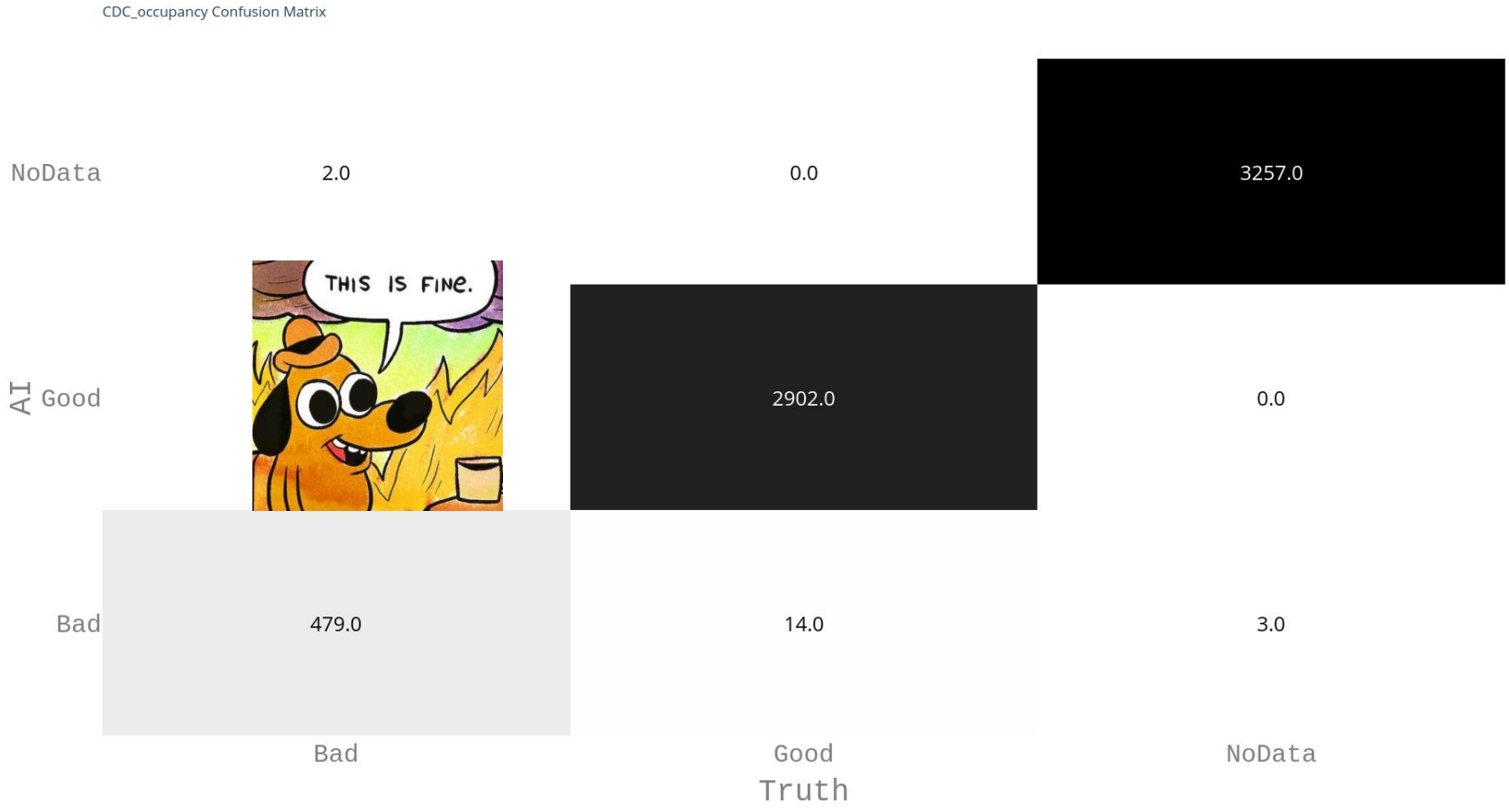
- Prediction performed by a script call
  - **3 modes**:
    - **Datum** (full path to image)
    - **Directory** (non-blocking attempts to analyze all data it finds)
    - **Crawl** (over all saved data in the DB)

- Writes and publishes (0MQ) **JSON report** including each label the model knows about and it's confidence in the label for each piece of data analyzed
  - Can be used in downstream processes
- Automatically captures examples for future training/analysis
  - "Bad" examples
  - Uncertain plots
  - Every Nth example

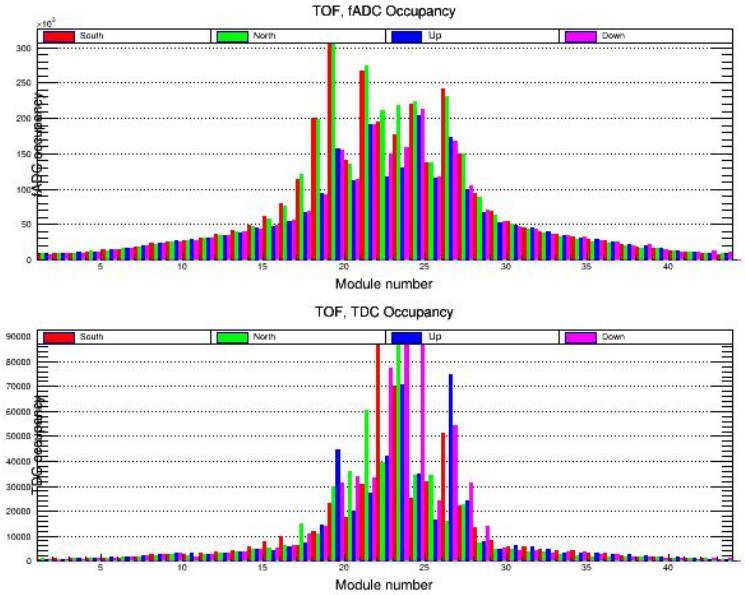
False Positives/Negatives

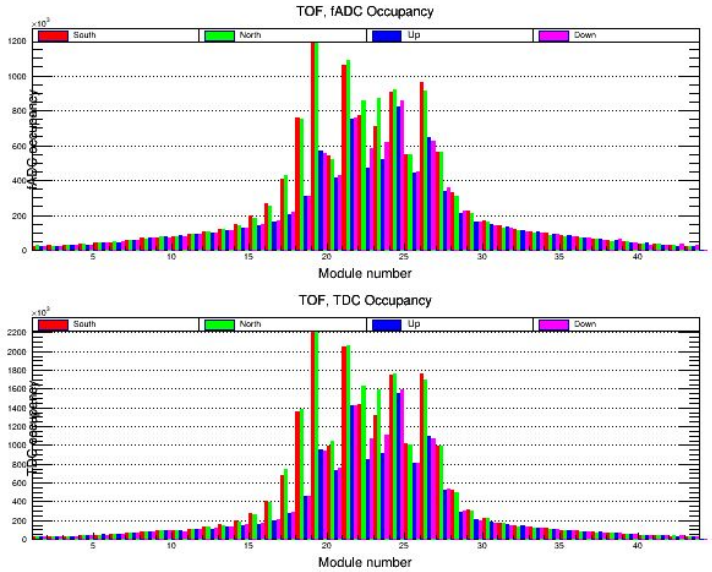


The Worst Case



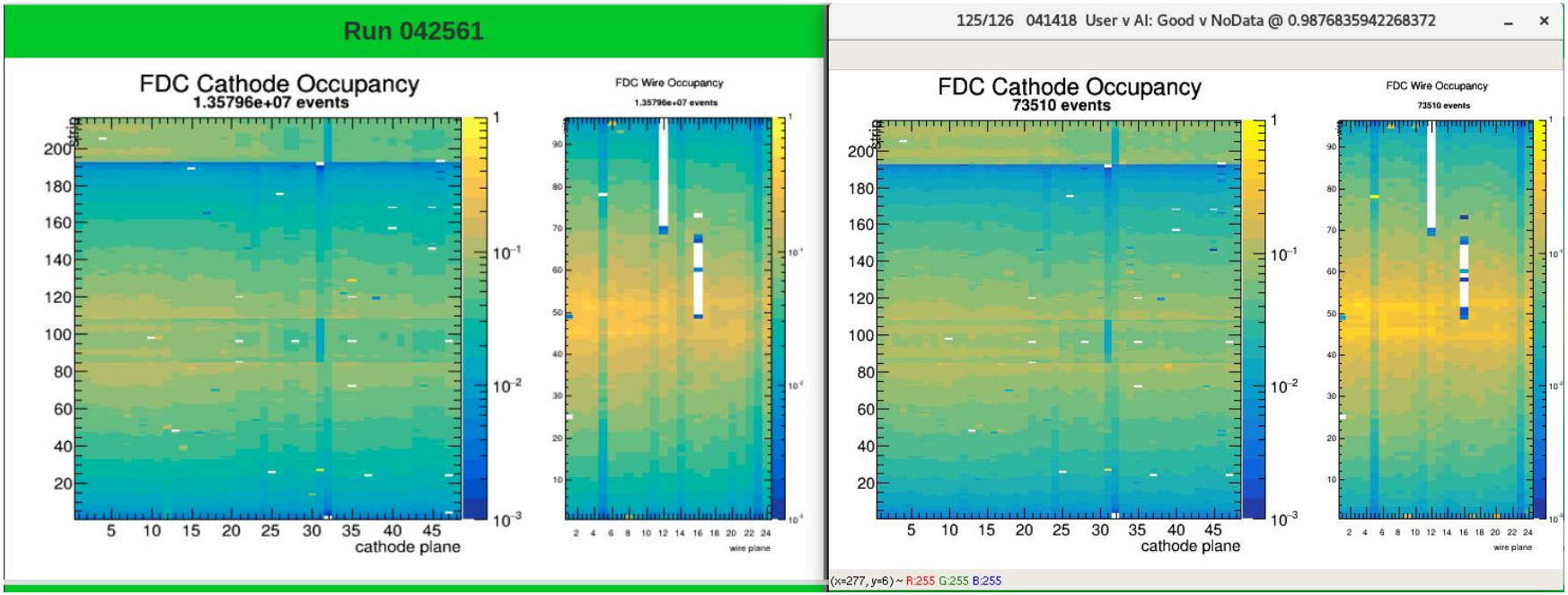
An Anecdote





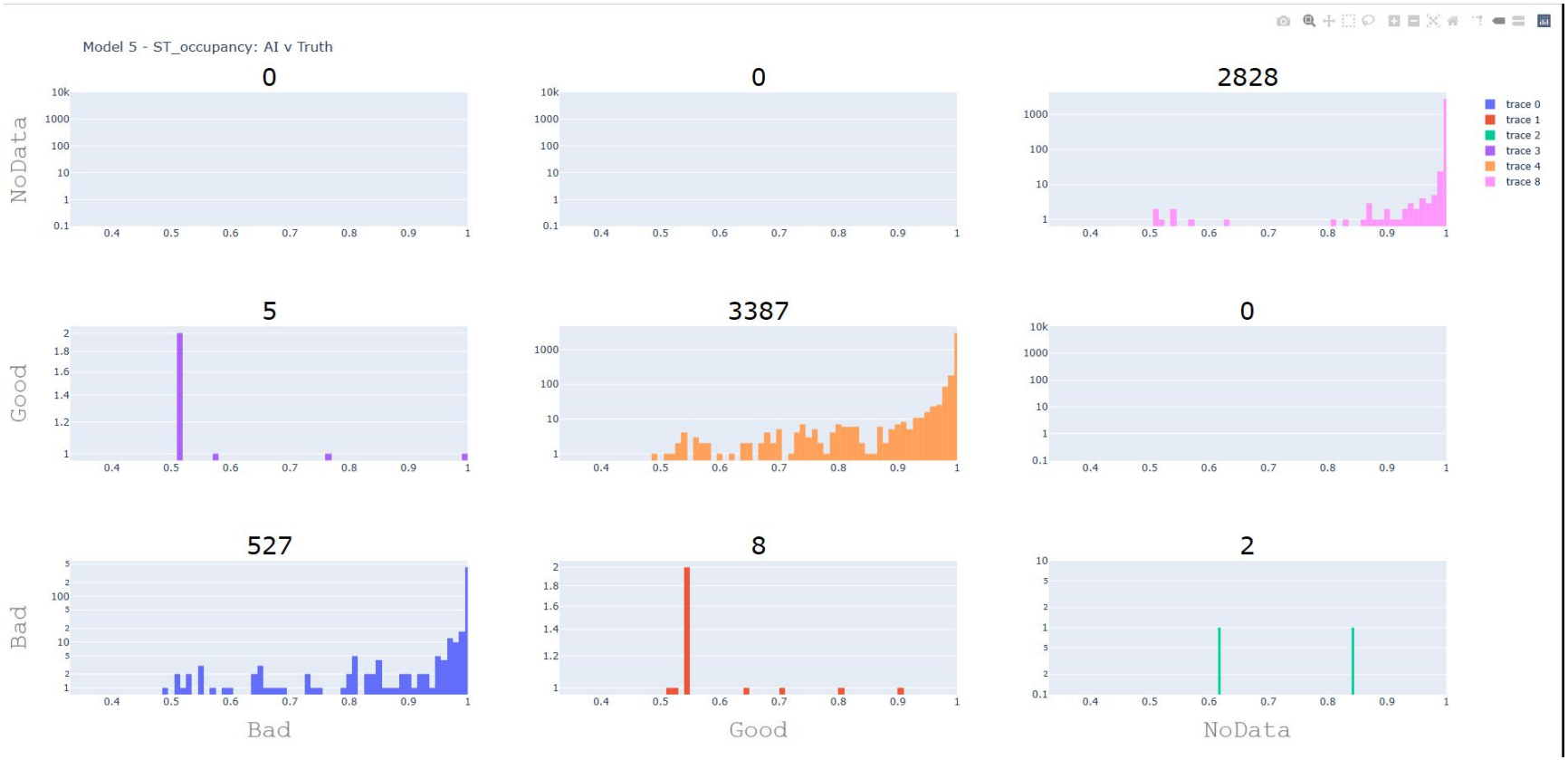
- Both of these look "good" at first glance (both initially labeled good)
  - The one on the left is actually bad (the A.I. caught it)
- A.I. seems to be able to look at subtle differences in shape and **maybe even "read" the y-axis**

Another Anecdote



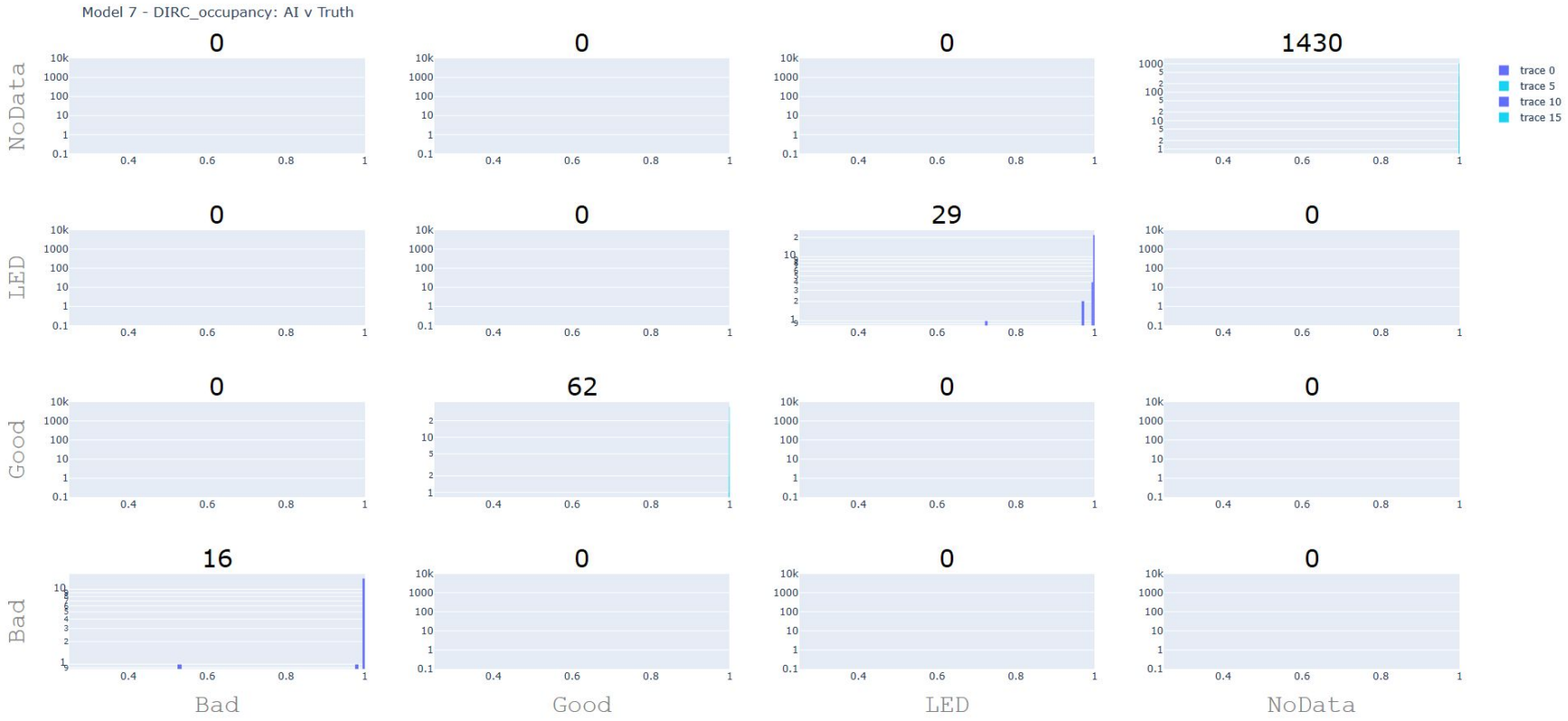
- The labeler was instructed by the detector expert to label any plot containing **fewer than 100k events as "NoData"**. This is one example of several in which the labeler labeled as "Good" and the **A.I. predicted "NoData"**...the true label given the number of events

Preliminary Results (Start Counter)



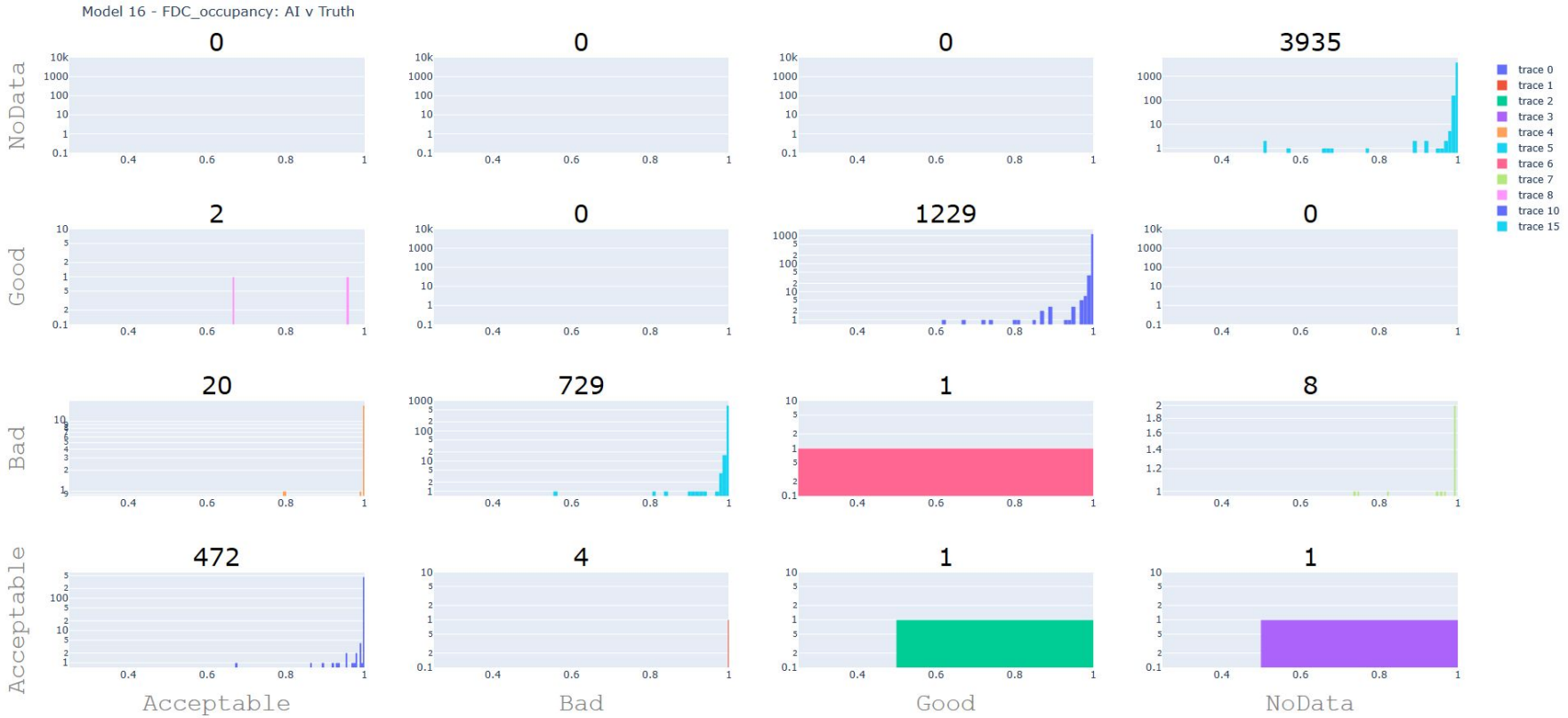
Preliminary Results (FCAL)

Preliminary Results (DIRC)

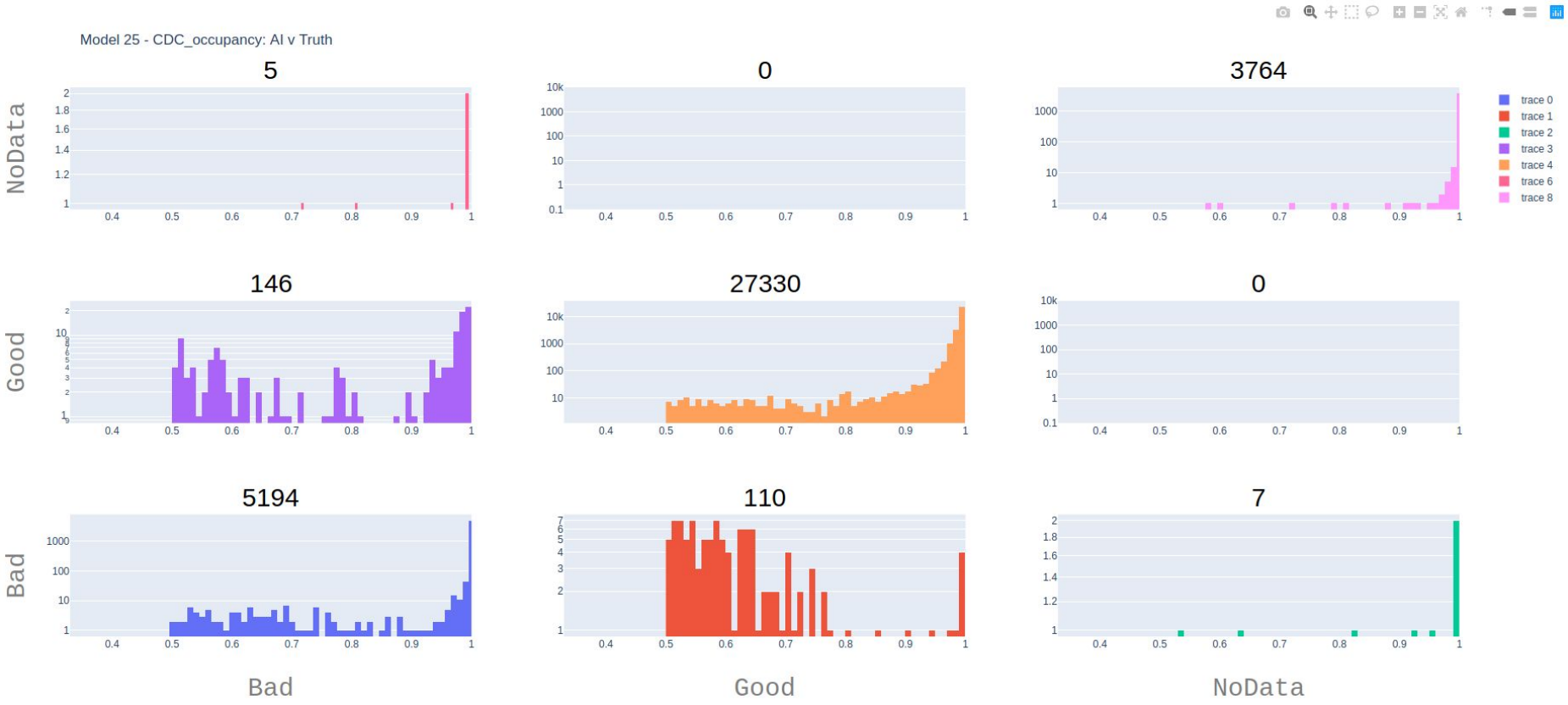


Preliminary Results (BCAL)

Preliminary Results (FDC)



Preliminary Results (CDC)



Preliminary Results (TOF)

Some Things I Noticed on the Way

- Tensorflow 1.14 and their partial bundling of keras is weird
  - I ended up having to use some of the tf.keras functions and some functions from keras standalone to get it to work
    - I've since started trying to move to TF 2.2 which is much better in this regard

- It looks like TF does some library loading when the model is loaded and more the first time predict is called for a model. Because I have many models that get used "on-the-fly" the predict times were terribly long (first predict in **4 seconds**)
  - Had to **run a prediction over an arbitrary image** on model load. Then all first predicts became fast.

```
def PreloadModels():
    then=int(time.time()*1000.0)
    print("Preloading Models")
    data_to_analyze_q="SELECT * FROM Plot_Types where Active_Model_ID IS NOT
    dbcursor.execute(data_to_analyze_q)
    data_to_analyze= dbcursor.fetchall()
    for d in data_to_analyze:
        print("Loading head for "+str(d["Name"]))
        chunked=-1
        if(d["IsChunked"] == 1):
            chunked=1
        LoadModel(d["Name"],d["FileType"],chunked,d["Active_Model_ID"])
        headkey=d["Name"]
        #print("BREAKFAST: "+str(chunked))
        if chunked==1:
            headkey=headkey+"_1"
        #print("BREAKFAST HEADKEY: "+headkey)
        Breakfast(headkey)
    now=int(time.time()*1000.0)
    print("Finished preloading all active models in: "+str(now-then)+" ms")
```

Hydra's Breakfast

Conclusion

- It is possible to employ computer vision techniques to handle many of the monitoring procedures

- A.I. is trivial to setup and get running
  - The devils are all over the details and can cause a lot of problems
    - Maintaining provenance
    - Interpretability
    - Validation
    - Optimizations

- Hydra has been deployed and already has taken some burden off shift crews. A big benefit during single person shift taking during COVID

- Lots of avenues for further research and improvements
  - Improved generalization via curation?
  - HPO
  - Increased analysis (new plots/data types)
  - Etc etc

Hot Off the Presses

- The prevailing wisdom is that, especially with deep learning, you need more data
  - This may be generically true but in certain situations less is more

- Label imbalance occurs when one or more labels are much greater in quantity than others
  - This can lead to poorly trained models as the optimum is heavily skewed
  - can input the smaller populations many times to “fake” a larger sample
    - this leads to longer training times
- Because the variance in “Bad” data is much greater than other labels we can undersample the larger samples to match the smaller sample
  - MUCH faster trainings for models that work just as well (or better)