mu2e Track Quality Selection in Python/Sklearn

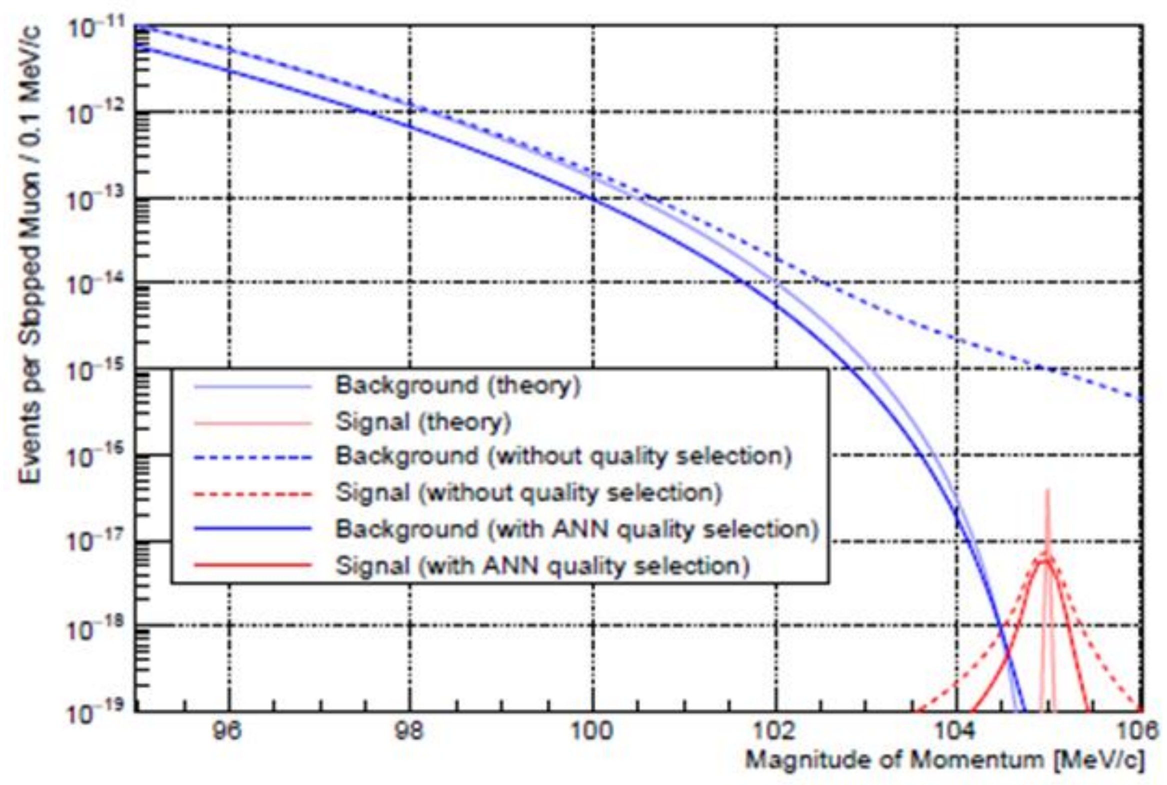
Scott Israel, Andrew Edmonds, Boston University

BOSTON UNIVERSITY

FERMILAB-POSTER-22-050-V

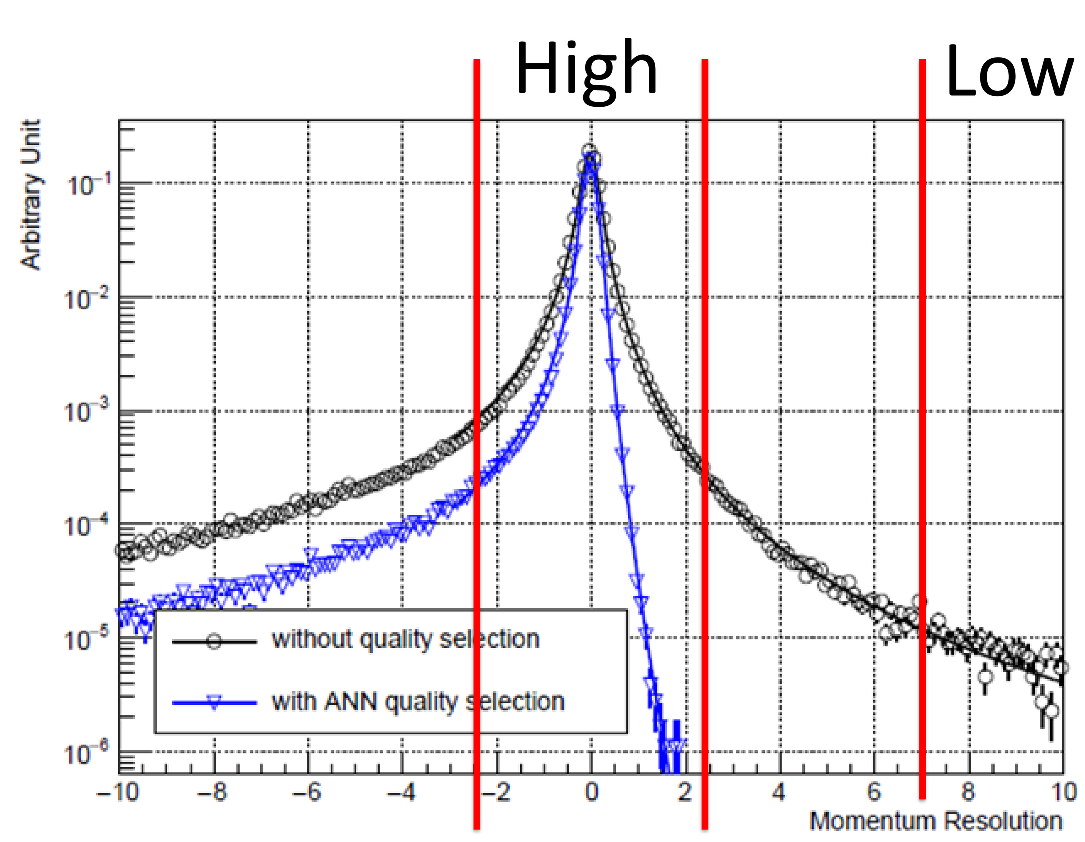
Track Quality

- Mu2e is looking for charged lepton flavor violation in the rare process: $N + \mu^- \to N + e^-$, on muonic aluminum down to a sensitivity of 10^{-16}
- Mu2e signal and background can be hard to distinguish
- Want to separate high/low quality measurements
- Select high quality measurements to boost signal sensitivity
- Used an ANN model in TMVA to do quality selection
- Issues
  - Neural networks are hard to diagnose with hidden layers
  - TMVA is ROOT/C++ based, lacks accessibility in documentation/tutorials vs Python
  - Students have more experience with Python, Python has a larger ML community



Example of the Decay in orbit (DIO) background burying the signal before quality selection, while the signal can be extracted without quality selection.

Goal is to make quality selected signal/background be as close to the theoretical signal/ background as possible.



ANN does a good job of filtering. How can we make this process better? Higher accuracy? More efficient model?

- Want to classify as high or low quality.
- High = |mom res| < 250 keV, label 1
- Low = mom res > 700 keV, label 0
- Ignore all other mom res, not used for training

Plan for Improvement

- Use a boosted decision tree
- Typically faster than neural networks
- Comparable performance

Model & Training Process

- Used same training features as Andy et al., training features = input variables

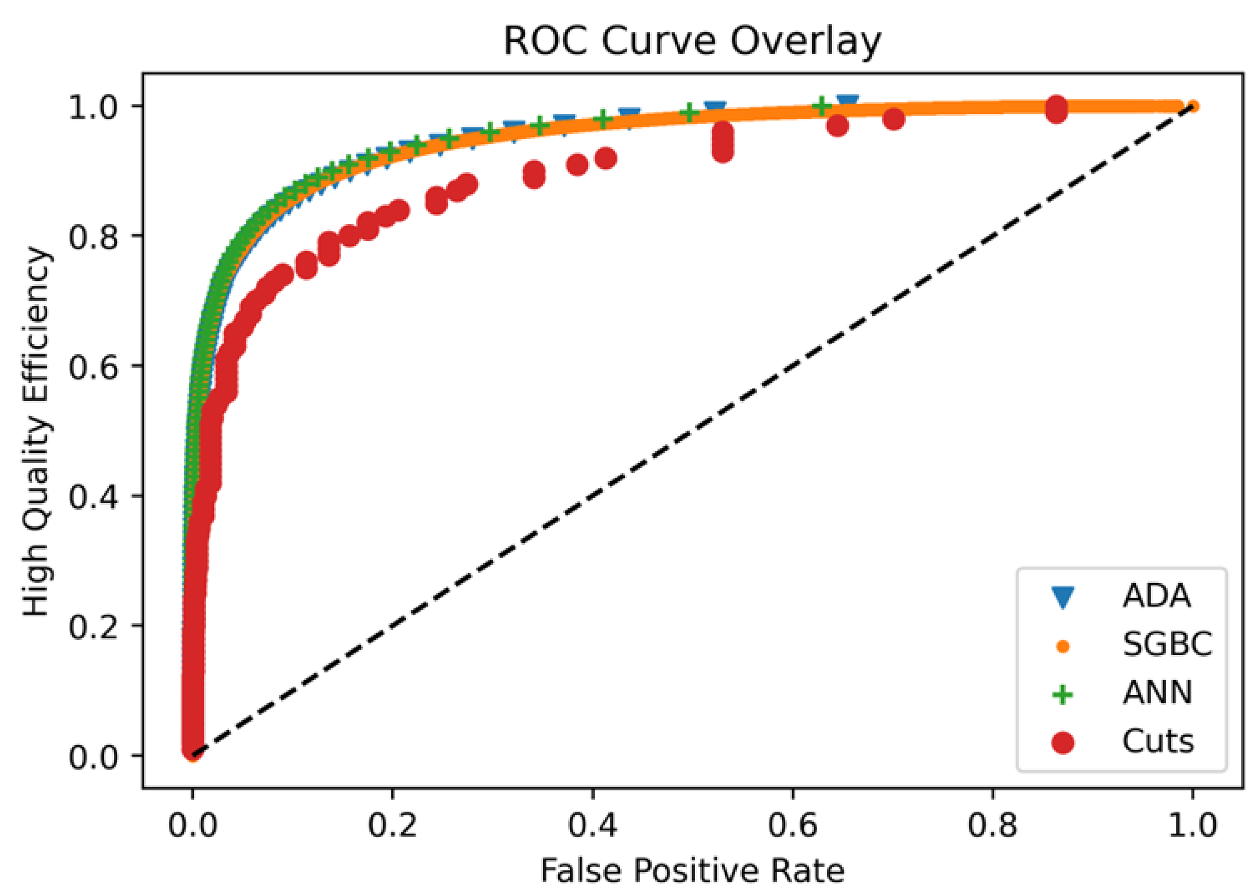
Input Variable	Brief Description
p_{err}	fit momentum error
n_{used}	number of used hits
$f_{expected}$	fraction of expected detector elements hit
f_{drift}	fraction of hits without a drift distance
f_{used}	fraction of used hits
t_{err}	fit time error
con	fit chi-square consistency

Model is a SGBC with max depth of 3, subsample 0.9, minimum samples per leaf of 150, and the number of total trees 850, same conditions as Andy et al. study in TMVA

	Python Sklearn	TMVA
F1 score	SGBC_BDT: 0.75	N/A
HQeff at 0.99 LQrej	SGBC_BDT: 59%, ADA_BDT: 57%	ADA_BDT: 57%, ANN: 60%
ROC_AUC score	SGBC_BDT: 0.95, ADA_BDT: 0.94	ADA_BDT: 0.95 , ANN: 0.95

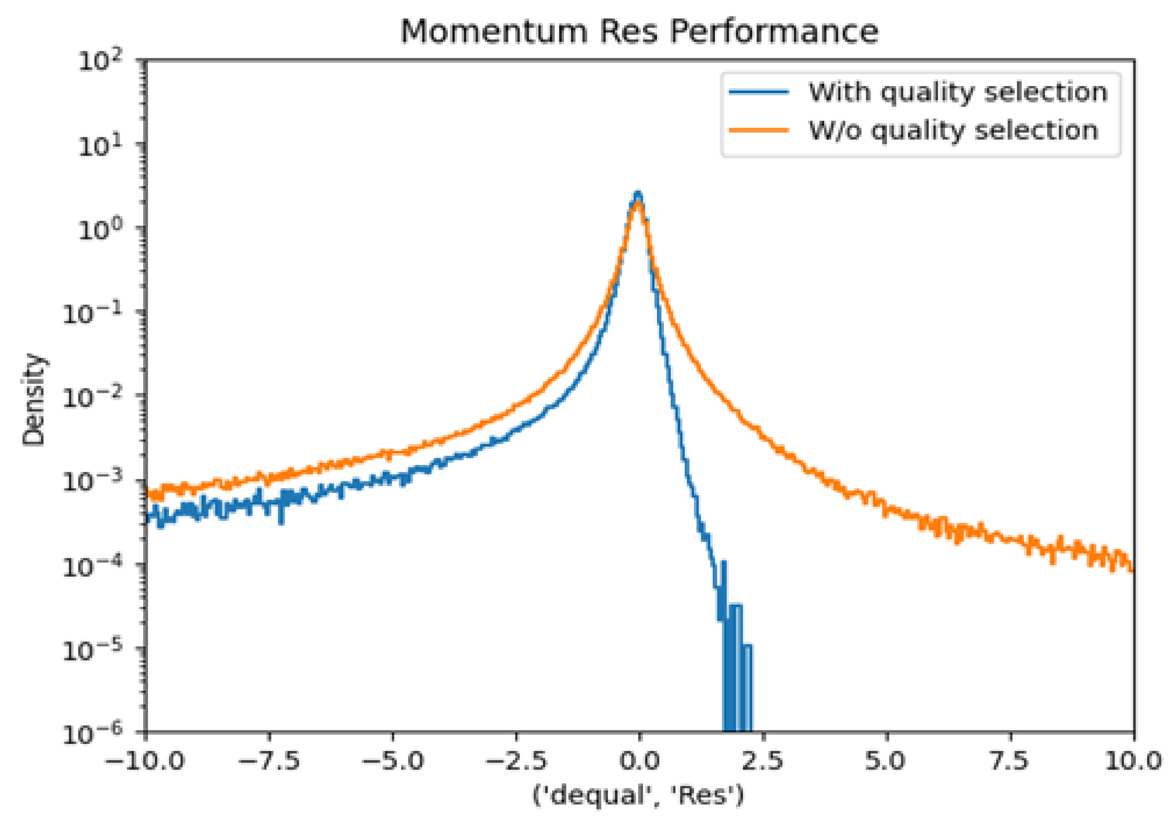
- SGBC = Stochastic gradient boosted classifier (decision tree)
- ADA = Adaptive boosted decision tree
- ANN = Artificial neural network

Model Training and Test Results



ROC curve with the results from the original TMVA ANN and BDT overlayed with the Sklearn ADA BDT and SGBC BDT compared to cuts in TMVA. The ML models all perform very well, particularly the ANN and SGBC

- Comparing original cuts, Ada BDT, ANN, and SGBC BDT
- Performance differences between ANN and SGBC are <1%
- **Main point**: Why use a SGBC over an ANN?
- Decision trees are simpler than neural networks, no hidden layers, easier to understand what the model is doing
- **Prediction time is <u>faster</u>**
  - TMVA ANN ~ 2.25e-5 s/entry
  - Sklearn SGBC BDT ~ 5.85e-6 s/entry



Besides some differences in normalization and scaling between TMVA and Sklearn, the Sklearn based SGBC filters just as well TMVA.

Comparing the Sklearn momentum resolution performance to the TMVA performance, they both filter the low-quality tracks out of the data. The noise at the bottom is a normalization difference between Sklearn and TMVA. It is present on TMVA but hidden below the lower xlim

Conclusions and Summary

- In machine learning, <u>a simpler model + good training data is preferred over a more complex model</u>
- SGBC performs equally as well as ANN and is simpler, more efficient, and easier to analyze in python
- Previous track quality classification track used an ANN and ADA BDT in TMVA
- New track quality classification uses a SGBC BDT in Sklearn
- Performance is comparable between SGBC and ANN, simplifies model considerably
- **<u>SGBC is quicker than ANN in training and predictions</u>**
- See DocDB 41505
- My code for the analysis: https://github.com/theittsco/mu2e/blob/master/ml/Final%20Versions/track_qual_training.py

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy