Machine Learning at for 2016 Vertexing Analysis

Matt Solt
SLAC National Accelerator Laboratory
HPS Analysis Workshop 2020
January 21, 2020

Introduction

- “Standard” analysis - use square cuts, then separated signal and background using z distribution
- Machine learning approach - more effectively separate prompt background from a displaced signal by utilizing all relevant vertexing/tracking information
- I show some very preliminary results with a random forest classifier for a single mass/epsilon value
- How do we make this a reality for 2016 Data?
  - I offer several options

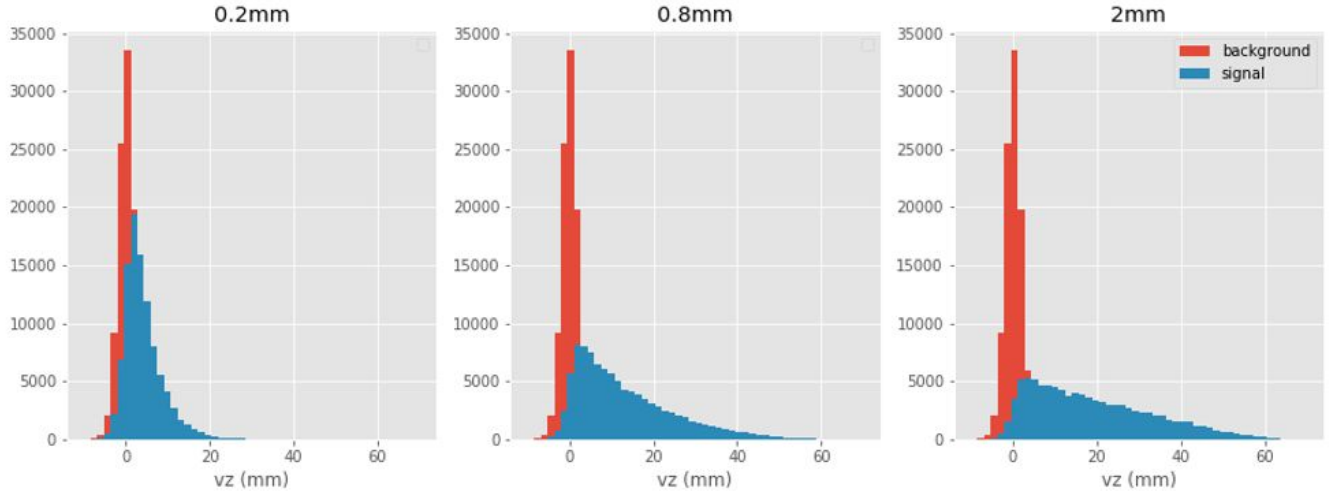
Classification Architectures

SLAC

- Neural Networks
  - I have shown this to work in the past
  - Too many hyperparameters (i.e. the parameters the user chooses) for this type of problem, not typically used for binary classification of tabular data with few features
  - Not interpretable
- Tree Ensembles
  - Random Forest
  - Gradient Boosted Machines (explore in the near future)
- Random Forest - focus on this for this talk
  - Fewer hyperparameters to tune
  - Simpler and somewhat interpretable

Over/Under Sampling

- At each mass, over/under sample signal events to get different distributions for different values of epsilon
- This avoids having to run MC for each epsilon (just each mass)



Train/Validate/Test Samples

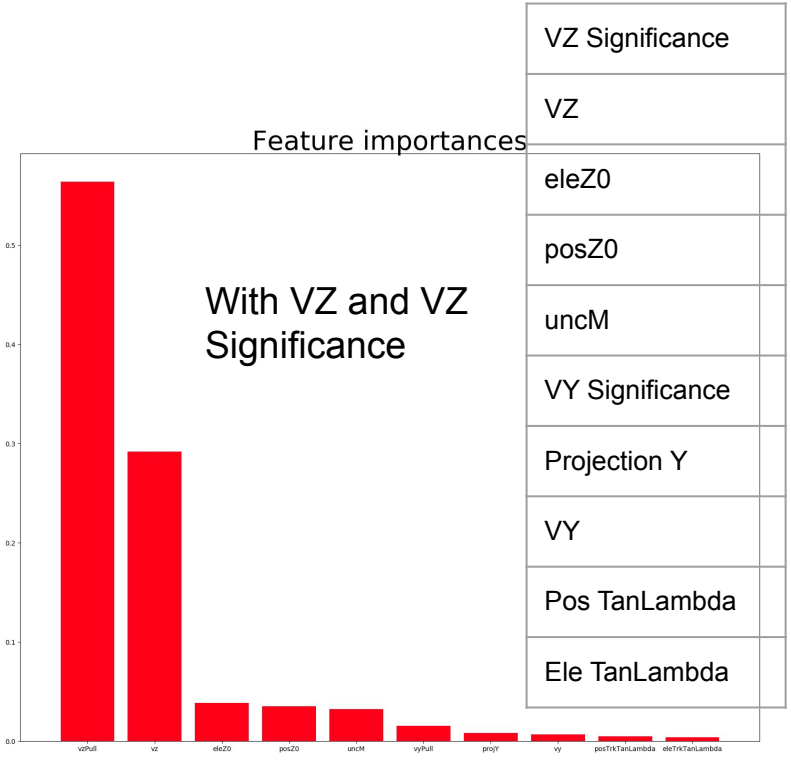
- This Dataset:
  - ⅓ * 100% tritrig-wab-beam for background
  - Ap-beam for signal (show results for 100 MeV, ctau = 0.8 mm) with over/under sampling
  - ~1.5 million background sample, ~0.1 million signal samples
  - Train/Test split is 67%/33%, respectively
- Future Dataset:
  - Use x3 tritrig for training/testing/validation
  - Use validation set for hyperparameter tuning
  - Train all masses and epsilons of interest

Training Data

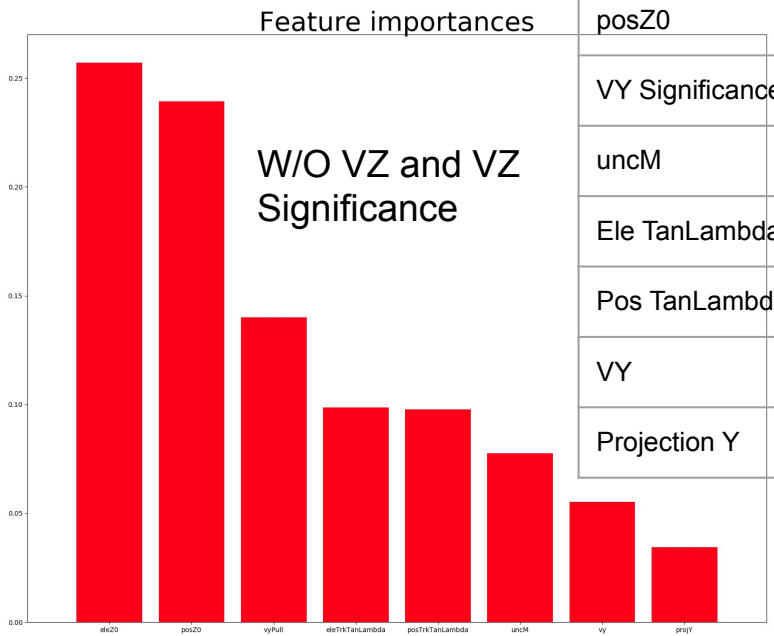
- These are older cuts, I haven't updated my cuts to match what I showed previously today

ele/pos has L1 & L2	Ele P < 1.725 GeV
ele/pos Track/Cluster Match Chisq < 10	V0 P < 2.645 GeV
Cluster Time Diff < 2 ns	V0 P > 1.84 GeV
Track Cluster Time Diff < 4 ns	90 MeV < uncM < 110 MeV
bscChisq < 10	ele/pos in opposite halves
ele/pos Track Chisq / dof < 6	

Feature Importances



- VZ Significance
- VZ
- eleZ0
- posZ0
- uncM
- VY Significance
- Projection Y
- VY
- Pos TanLambda
- Ele TanLambda



- eleZ0
- posZ0
- VY Significance
- uncM
- Ele TanLambda
- Pos TanLambda
- VY
- Projection Y

Feature Selection

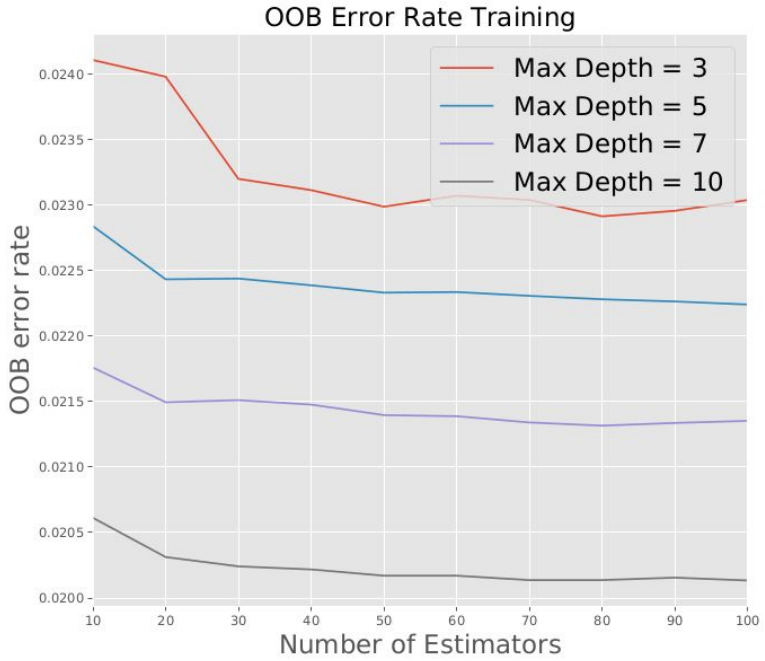
- Focus on Variables in y and z directions
- 1st set with VZ and VZ significance
- 2nd set without VZ and VZ significance, but with VZ Error
- I only have results for the first case so far

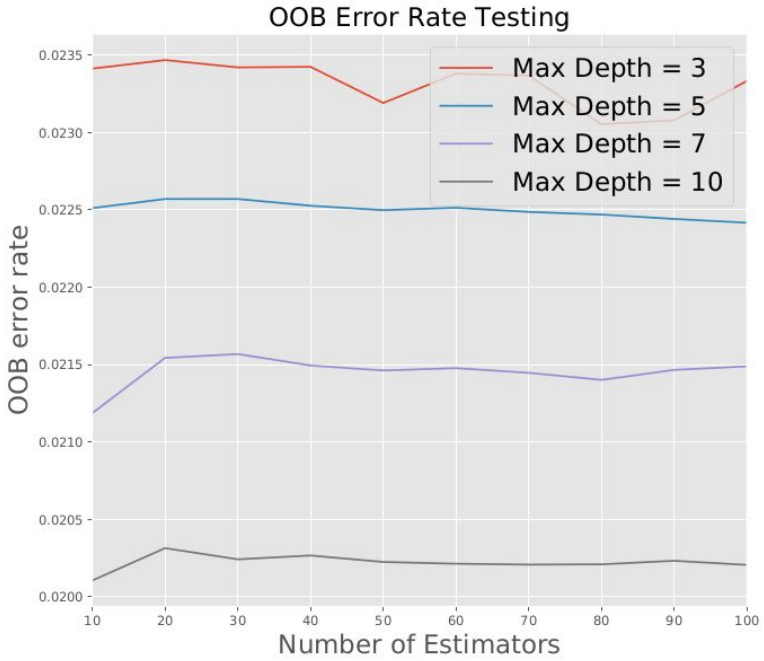
VZ Significance
VZ
eleZ0
posZ0
uncM
VY Significance
Projection Y
VY
Pos TanLambda
Ele TanLambda

VZ Error
eleZ0
posZ0
uncM
VY Significance
Projection Y
VY
Pos TanLambda
Ele TanLambda

Hyperparameter Tuning

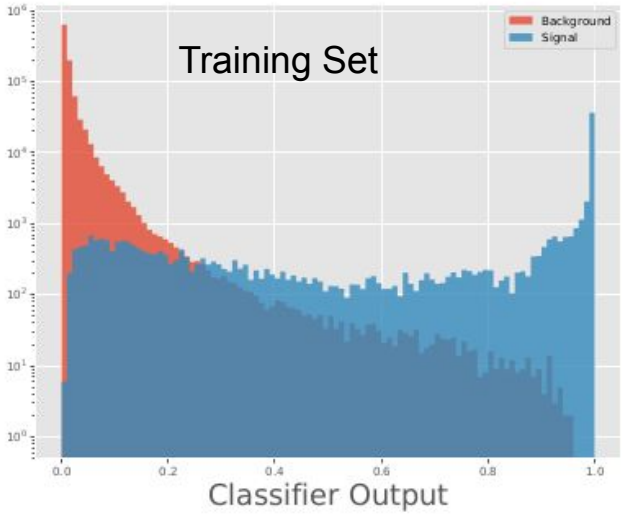
- Example of Hyperparameter tuning: number of trees, max depth of trees
- There are a few other hyperparameters to tune





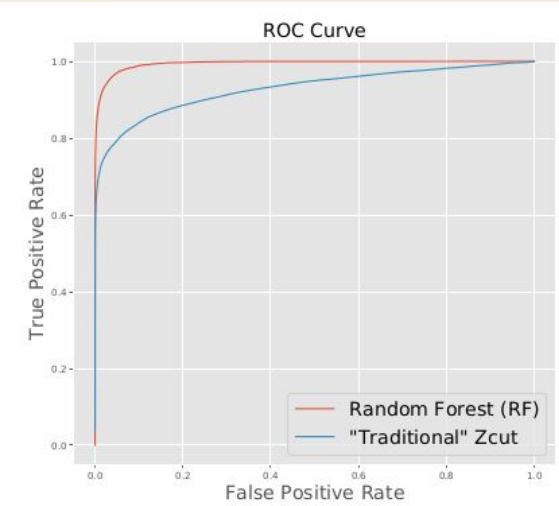
Training Classifier Output

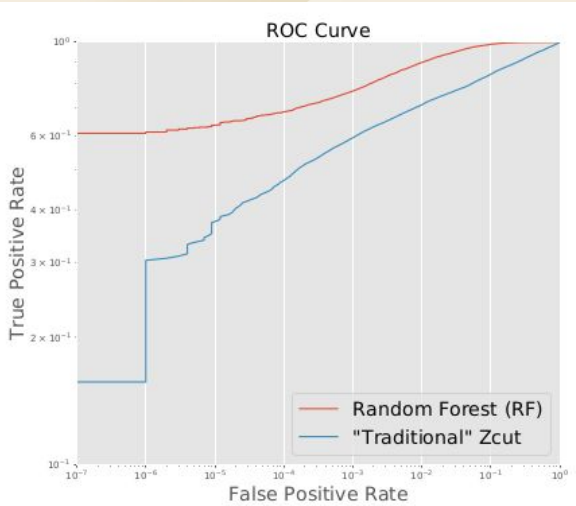
- Classifier outputs a score between 0 (more background-like) to 1 (more signal-like)
- Training (left). Testing (right)

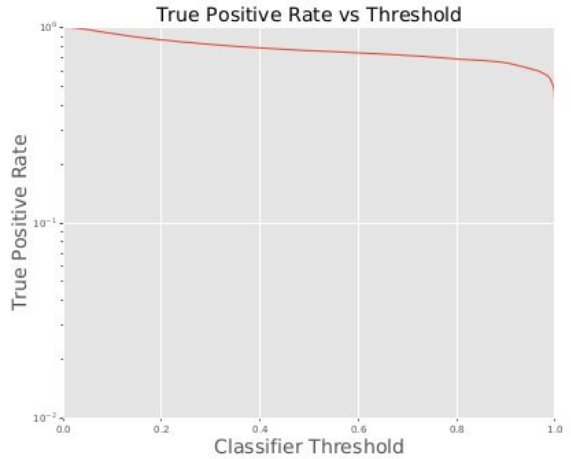


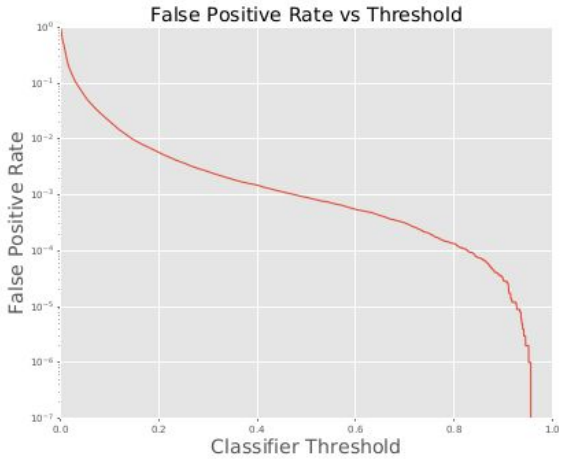
Training ROC Curves

- Training ROC curves
- Compares ROC curves from random forest to traditional zcut method

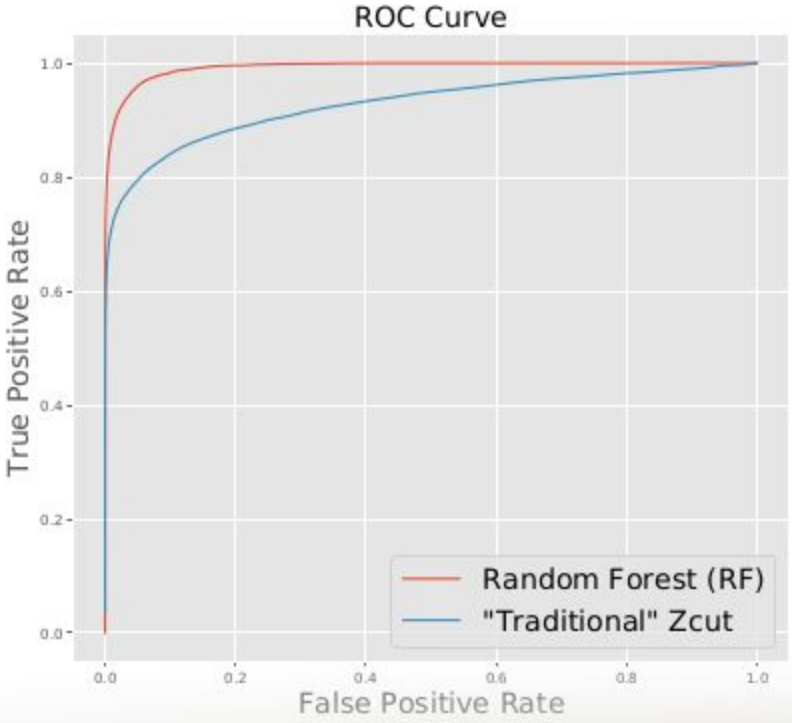


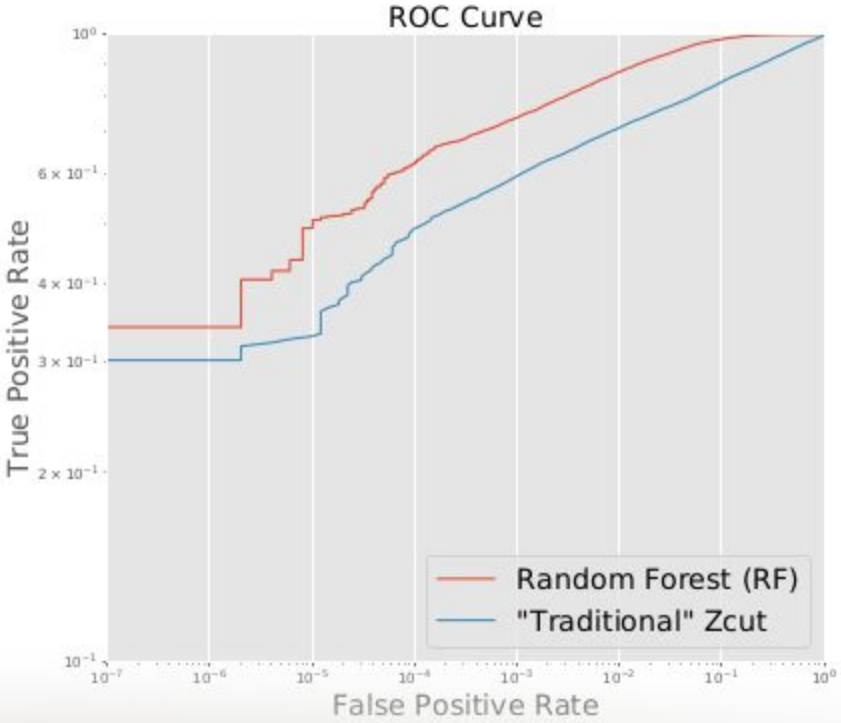






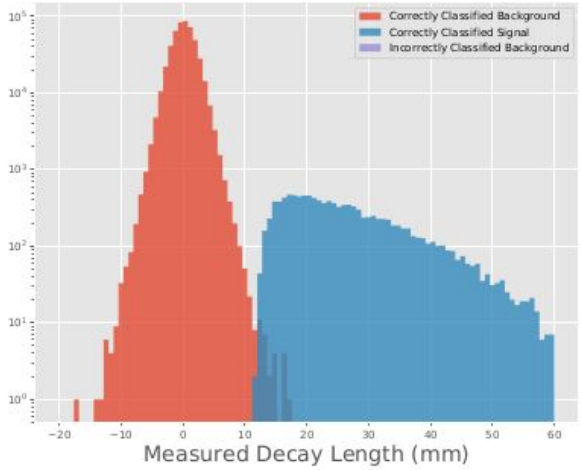
Testing ROC Curve

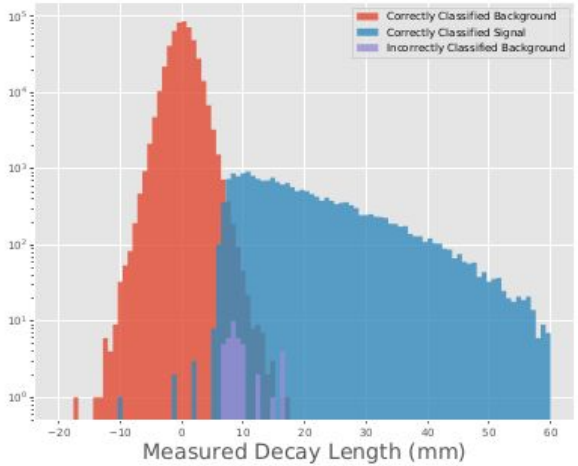




Testing Results

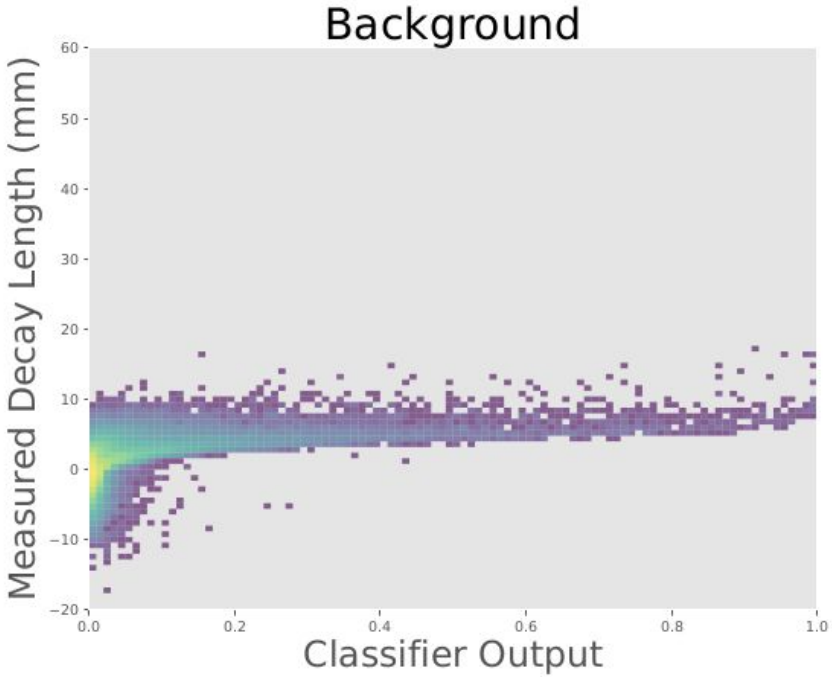
- Left: Testing set results with classifier cut determined by *testing* ROC curve
- Right: Testing set results with classifier cut determined by training ROC curve
- I am probably overfitting the training set right now...





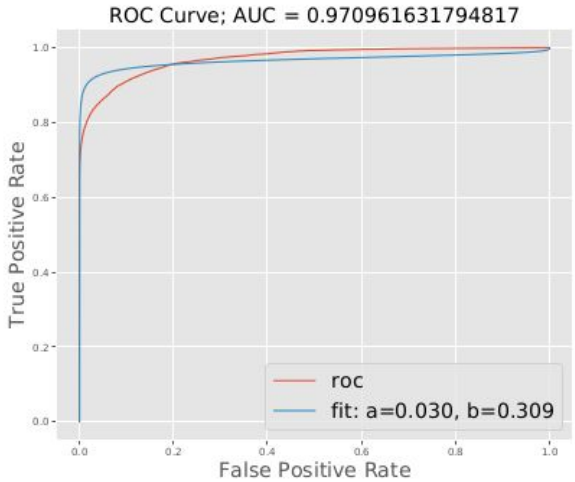
Testing Set Results

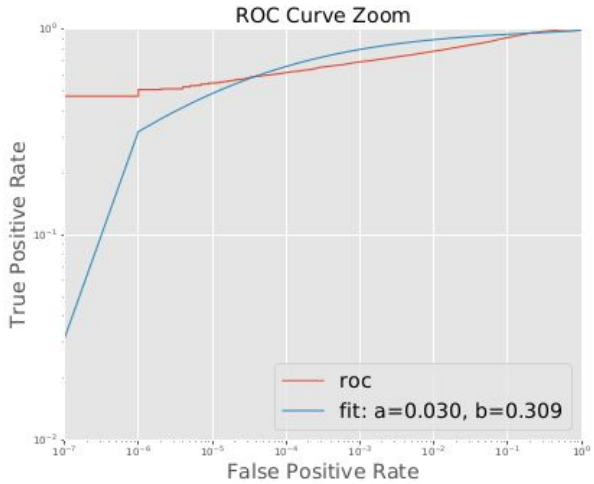
- Z vs output
- Rejects many events on the tails of the VZ distribution



Fitting ROC Curves in Training Set

- Fit the ROC curve to predict (from training set) the threshold for which you expect 0.5 background (in test set)
- First attempt to fit… doesn't work so far, need to explore





Making this Possible for 2016 Vertex Analysis

- Use full tritrig MC sample for training
- Use validation set for hyperparameter tuning
- Train the full mass/epsilon ranges
- Interpolate between masses
- Estimate Systematics
- Obtain approval for this approach
- Possible uses for 2016 vertexing analysis:
  - Just for fun (i.e. it goes in my thesis, but not part of the published analysis)
  - Use to reject background
  - Use the classifier output to set limit