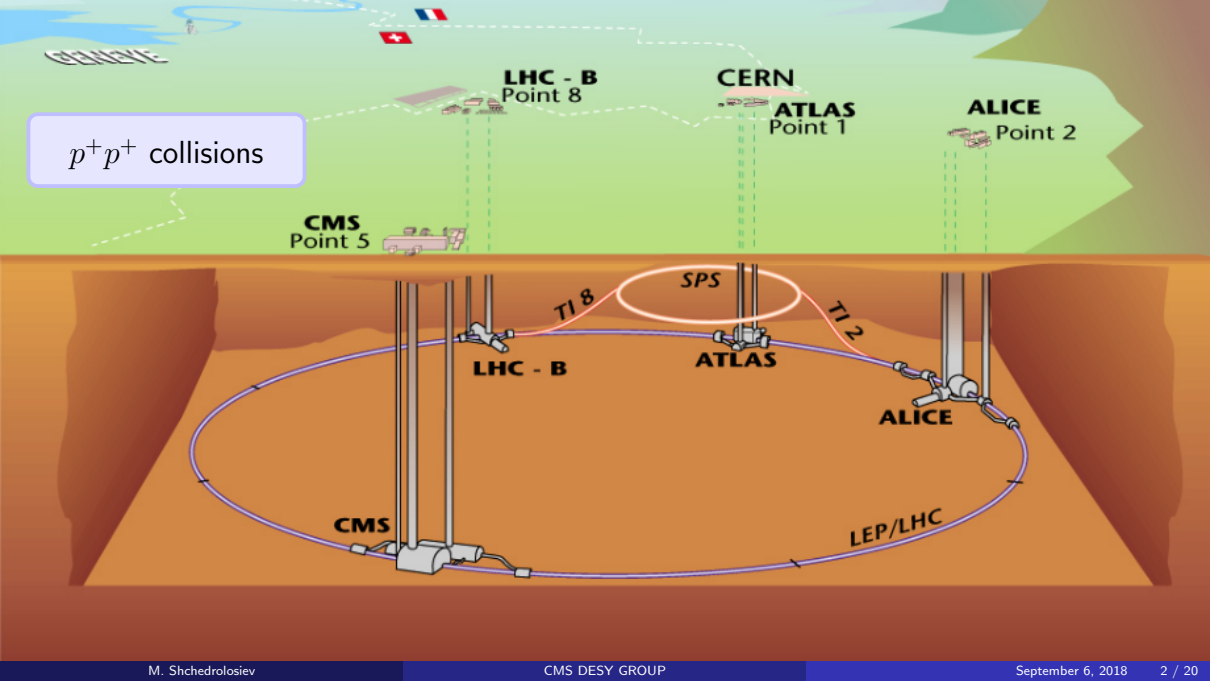
Direct optimization of the discovery significance for Machine Learning Analysis in CMS SUSY stop search

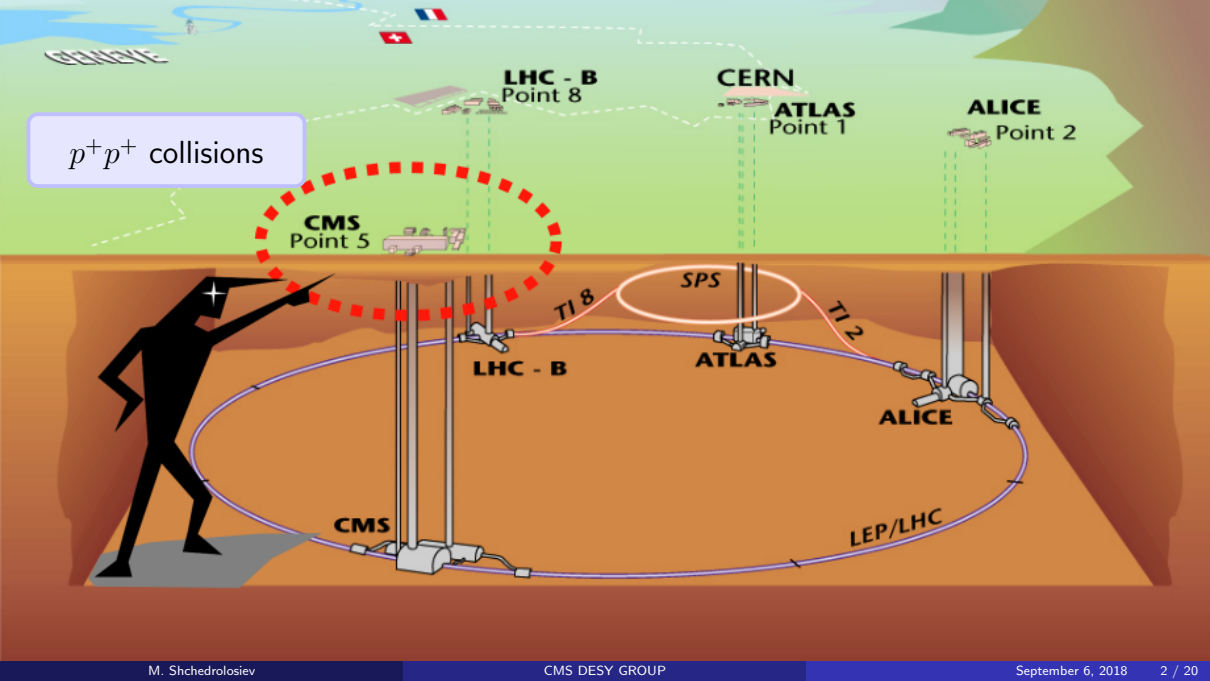
M.Shchedrolosiev

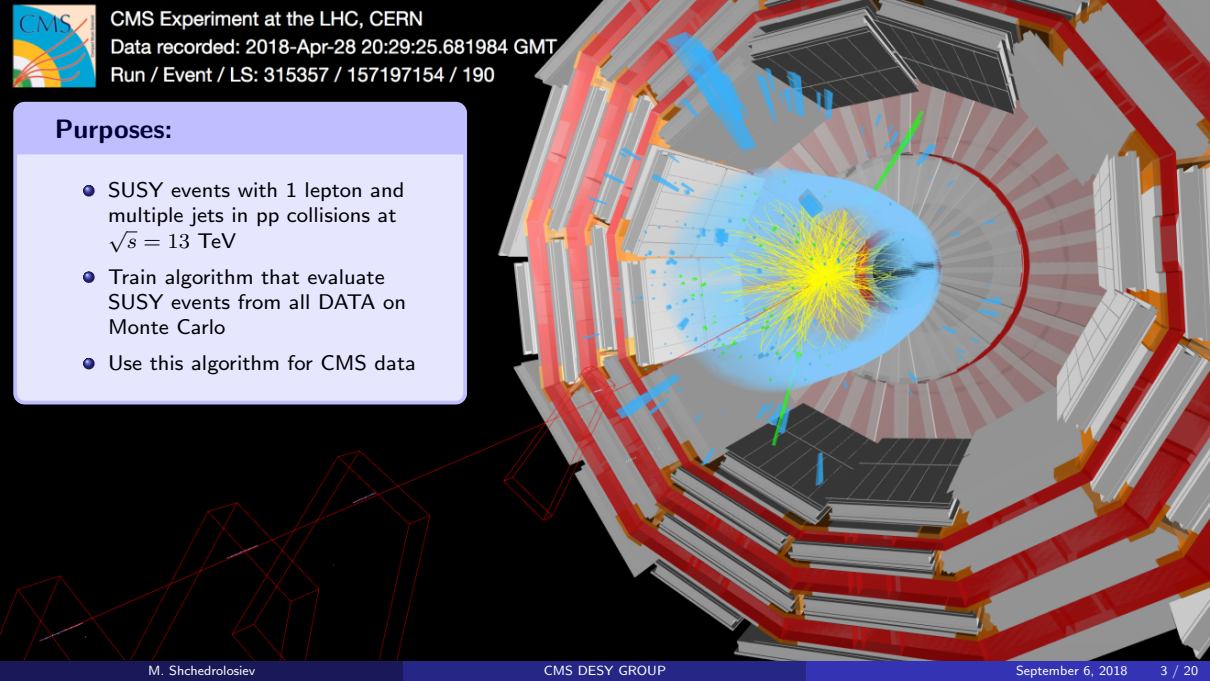
Content:

- CMS experiment
- Direct optimization of the discovery significance
- XGBoost (Boosted decision tree approach)
- Stop SUSY model Monte Carlo
- Results

Supervised by: A.Elwood, D.Krücker, I.Melzer-Pellmann, O.Turkot
Thanks for contribution to: C.Contreras


GENEVE
LHC - B
Point 8
CERN
ATLAS
Point 1
ALICE
Point 2
CMS
Point 5
p^+p^+ collisions
TI 8
SPS
TI 2
LHC - B
ATLAS
ALICE
CMS
LEP/LHC


GENEVE
LHC - B
Point 8
CERN
ATLAS
Point 1
ALICE
Point 2
p^+p^+ collisions
CMS
Point 5
TI 8
SPS
TI 2
LHC - B
ATLAS
ALICE
CMS
LEP/LHC



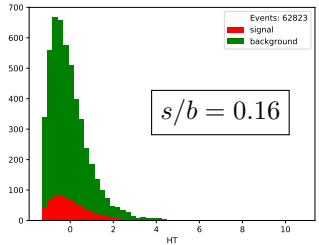
Purposes:

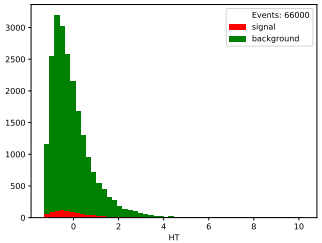
- SUSY events with 1 lepton and multiple jets in pp collisions at $\sqrt{s} = 13$ TeV
- Train algorithm that evaluate SUSY events from all DATA on Monte Carlo
- Use this algorithm for CMS data

Direct optimization of the discovery significance

- When searching for new physics the most important is the significance of signal counts over background counts, but purity of the background classification is not very important

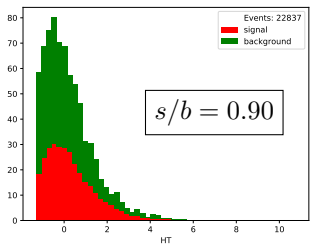
Standard ML approach
Signal prediction





input data

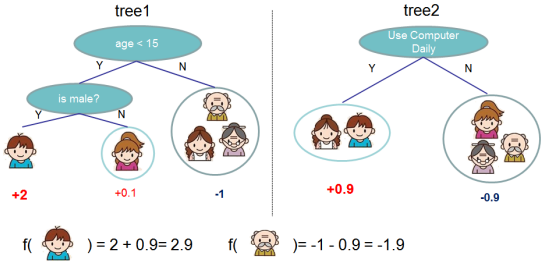
with a HEP approach
Signal prediction



- To train Machine Learning algorithms at HEP approach we want to directly maximize discovery significance, not accuracy or ROC curve area, to get a sample where the signal dominates in signal prediction

XGBoost (Boosted decision tree approach)

Example: does person play PC games?



- Prediction is sum of scores predicted by each of the tree

- Gradient Tree Boosting :
  - $\hat{y}_i^{(t)}$ is prediction of the i-th instance at the t-th iteration, Ω - penalizes the complexity of the model
  - XGBoost uses second order approximation
  - Additive Training (Boosting)
  - Minimize every next tree f_t

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i)] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the gradient statistics on the loss function.

Direct optimization of the discovery significance[1]

- The standard approach is to maximize accuracy through minimizing Binary cross entropy:

$$C = -\frac{1}{n}\sum_{x}[y^{true}\ln y^{pred} + (1-y^{true})\ln(1-y^{pred})]$$

- equivalent to minimize logarithmic likelihood for binomial model

- To train neural networks in HEP approach, one can design a loss function based around the direct optimization of the discovery significance, for instance maximize Asimov discovery significance (minimize loss $l_{Asimov} = 1/Z_A$):

$$Z_A = \sqrt{2((s+b)\ln[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}] - \frac{b^2}{\sigma_b^2}\ln[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}]} \to \frac{s}{\sqrt{s+b}}$$

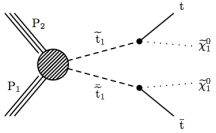
  - **s** - correctly classified signal events
  - **b** - incorrectly classified background events
  - σ - systematic uncertainty

$$s = W_s \sum_{i}^{N_{batch}} y_i^{pred} \times y_i^{true}, \qquad b = W_b \sum_{i}^{N_{batch}} y_i^{pred} \times (1-y_i^{true}),$$

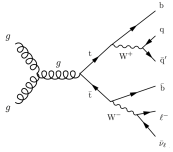
[1] arXiv:1007.1727v3

Stop SUSY model MC[2]

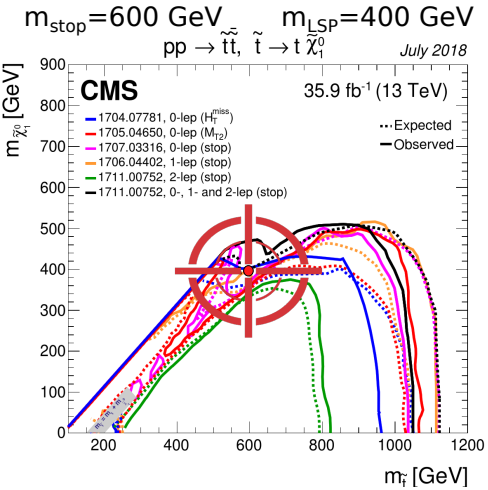
- To test this approach look at the stop SUSY model that is close to the edge of exclusion at $30 fb^{-1}$ of 13 TeV LHC data
- Consider top/anti-top quark pairs as the background
- Taken 1M events of signal and background with Pythia and Delphes with basic selection criteria: 1 lepton pT$\geq$40 GeV, 4 jets pT$\geq$30 GeV, at least 1 b-tagged jet



Signal

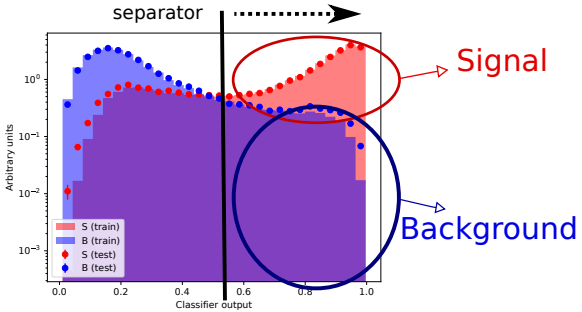


Background



[2]https://indico.desy.de/indico/event/21116/contribution/0/material/slides/0.pdf

XGBoost (Classifier output)



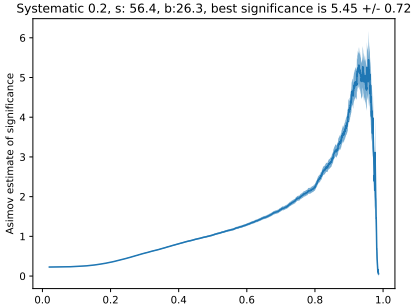
- Take events that were classified with value more than some fixed value of separator
- Calculate how much signal and background there
- Calculate value of functional that we want to optimize (for instance: Asimov significance)

$$Z_A = \sqrt{2((s+b)\ln[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}] - \frac{b^2}{\sigma_b^2}\ln[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}]}$$

Note that:

3σ - observation

5σ - discovery



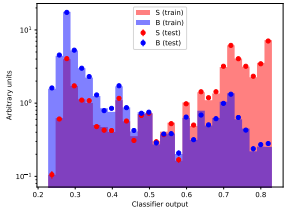
XGBoost out-of-box (Asimov estimate scorer, Z_A)

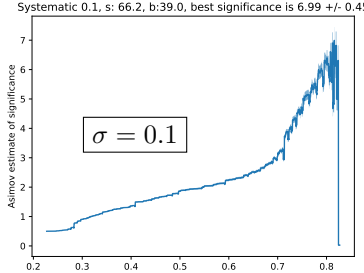
- Much worse comparing to the neural network[2]

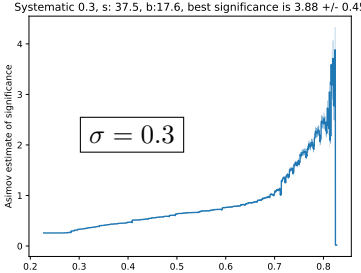
σ	0.1	0.3	0.5
Asimov NN	10.7	6.8	4.8
Asimov XGBoost	7.0	3.9	2.6

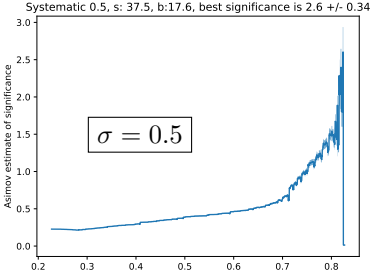
[2] arXiv:1806.00322

- Has to be tuned basing on Asimov significance
- Early stopping is used basing on Asimov scorer



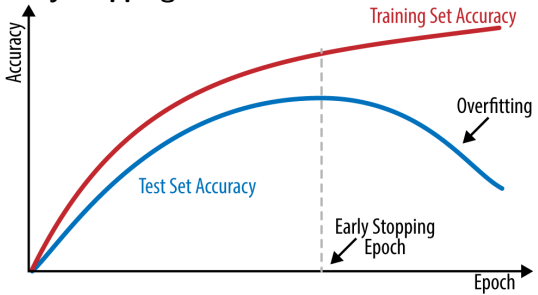






XGBoost (tuning and implementation of objective loss function)

Early stopping:



- Asimov score as a metrics for early stopping procedure
- To avoid an error in early stopping continuously differentiable function of Asimov function was used

XGBoost objective function

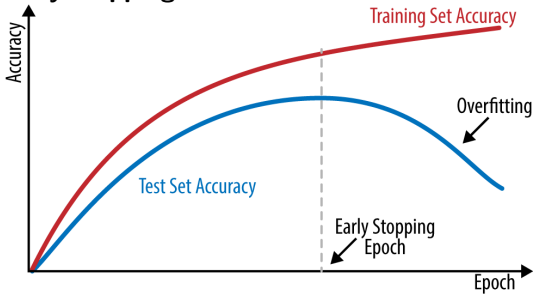
$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(_i)] + \Omega(f_t)$$

where we define loss function as: $l_{Asimov} = 1/Z_A$ and gradient:

$$g_i = \frac{\partial l(y_i, \hat{y}^{(t-1)})}{\partial \hat{y}^{(t-1)}}$$

XGBoost (tuning and implementation of objective loss function)

Early stopping:



- Asimov score as a metrics for early stopping procedure
- To avoid an error in early stopping continuously differentiable function of Asimov function was used

XGBoost objective function

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(_i)] + \Omega(f_t)$$

where we define loss function as: $l_{Asimov} = 1/Z_A$ and gradient:

$$g_i = \frac{\partial l(y_i, \hat{y}^{(t-1)})}{\partial \hat{y}^{(t-1)}}$$

$$g_i = -Z_A^{-2}\left(\frac{\partial Z_A}{\partial s} W_s y_i + \frac{\partial Z_A}{\partial b} W_b (1 - y_i)\right)$$

XGBoost (tuning and implementation of objective loss function)

Binary cross entropy as a minimization function in fit

$$C = -\frac{1}{n}\sum_x [y \ln a + (1-y)\ln(1-a)]$$

Asimov significance as a minimization function in fit

$$Z_A = [2((s+b)\ln[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}] - \frac{b^2}{\sigma_b^2}\ln[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}]]^{1/2}$$

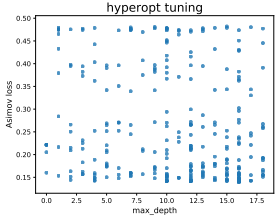
Tuning

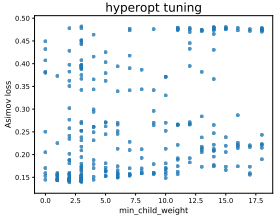
- Changed and adjust separation facet for predicting of class
- Used Asimov score as a metrics for early stopping approach
- Used Asimov score as a metrics for a hyperparameter tuning

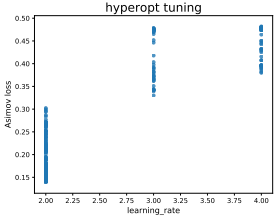
Goal is to compare both after tuning!

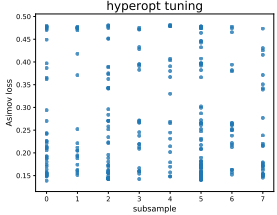
XGBoost (Hyperparameter tuning)

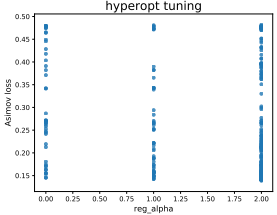
- Put Asimov loss function as an evaluation function for hyper-parameter tuning $l_{Asimov} = 1/Z_A$. Find optimal parameters for XGBoost Classifier:

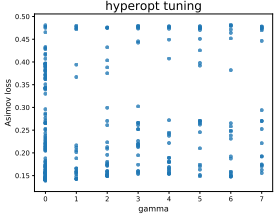




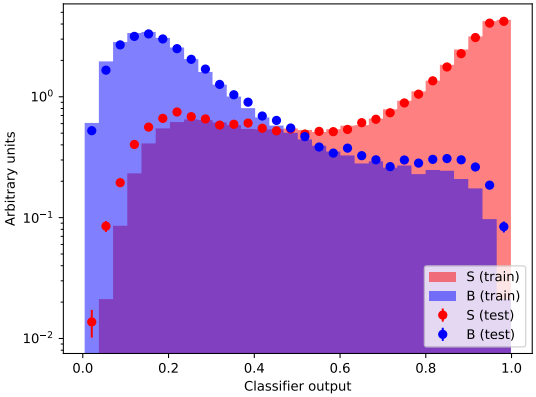




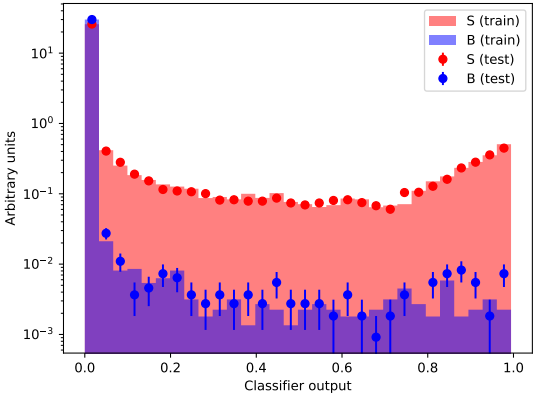




XGBoost (Train and Test set comparison)



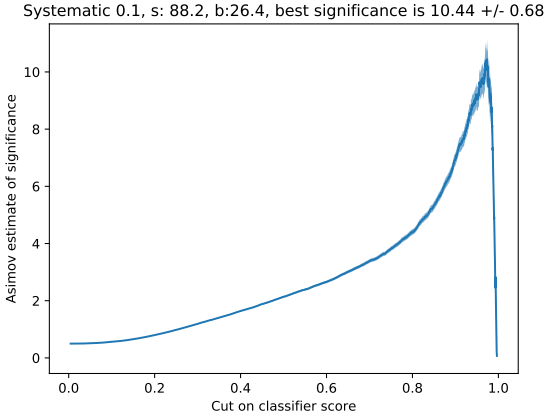
Binary cross entropy loss function

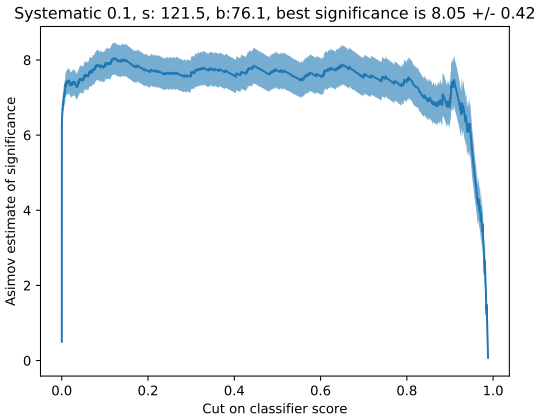


Asimov loss function

Asimov significance demonstrates a good separation of signal and background in a wide range of probability values

XGBoost ($\sigma = 0.1$)



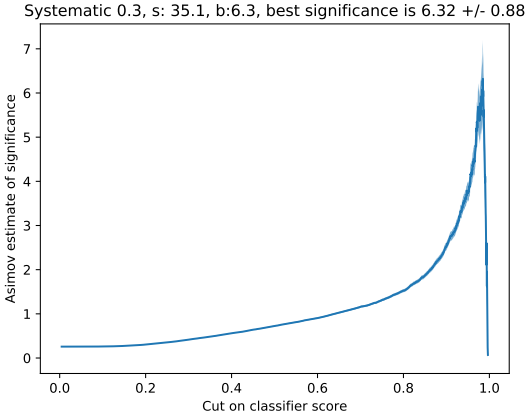


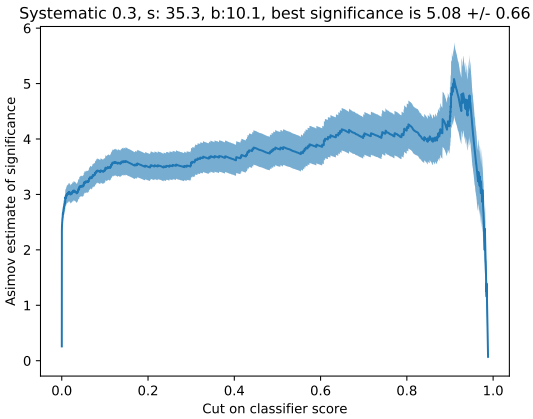
Binary cross entropy loss function

Asimov loss function

Binary cross entropy demonstrates better result than Asimov significance as an objective function for small uncertainty

XGBoost ($\sigma = 0.3$)



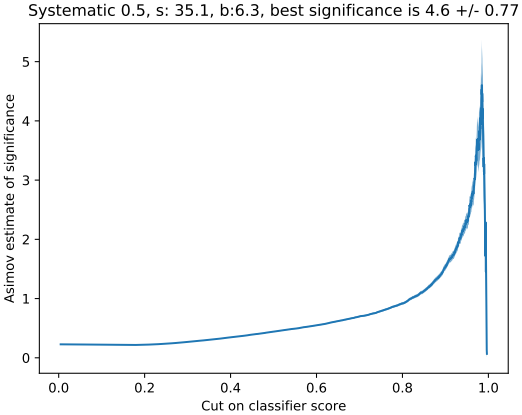


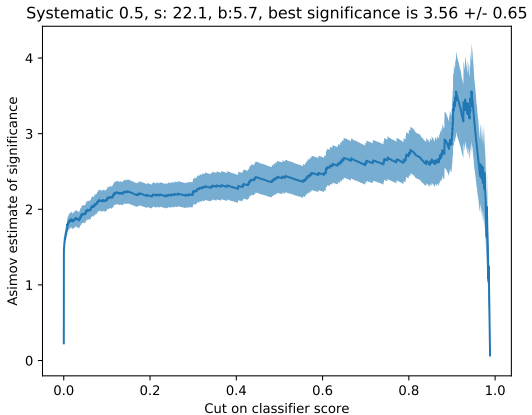
Binary cross entropy loss function

Asimov loss function

The difference between Binary cross entropy and Asimov significance diminishes with the increase of uncertainty

XGBoost ($\sigma = 0.5$)





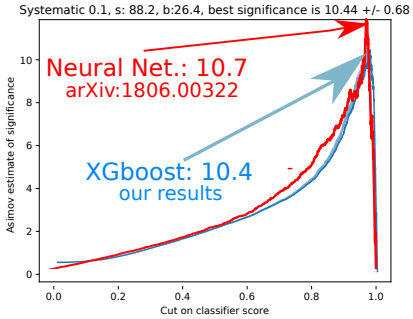
Binary cross entropy loss function

Asimov loss function

Binary cross entropy and Asimov loss are comparable within uncertainty ranges for $\sigma = 0.5$

Summary

- Implemented XGBoost classifier in framefork for the SUSY analysis
  https://github.com/shedprog/hepML
- Implemented Asimov loss as an objective function and evaluating metrics for an early stopping and for hyperparameter tuning
- Showed that tuned XGBoost with the default binary cross entropy performs almost as good as the neural network
- Showed that using of Asimov loss as an objective function lead to small improvement (but very close to binary cross entropy for high systematic)




THANK YOU!