

Entrenamiento BTD para identificar signal de background con 4 leptones

Noé Camacho



4lep 13 TeV data, formato root

Table 4: Description of the 13 TeV ATLAS Open Data branches and variables.

Tuple branch name	C++ type	Variable description
runNumber	int	number uniquely identifying ATLAS data-taking run
eventNumber	int	event number and run number combined uniquely identifies event
channelNumber	int	number uniquely identifying ATLAS simulated dataset
mcWeight	float	weight of a simulated event
XSection	float	total cross-section, including filter efficiency and higher-order correction factor
SumWeights	float	generated sum of weights for MC process
scaleFactor_PILEUP	float	scale-factor for pileup reweighting
scaleFactor_ELE	float	scale-factor for electron efficiency
scaleFactor_MUON	float	scale-factor for muon efficiency
scaleFactor_PHOTON	float	scale-factor for photon efficiency
scaleFactor_TAU	float	scale-factor for tau efficiency
scaleFactor_BTAG	float	scale-factor for b -tagging algorithm @ 70% efficiency
scaleFactor_LepTRIGGER	float	scale-factor for lepton triggers
scaleFactor_PhotonTRIGGER	float	scale-factor for photon triggers
trigE	bool	boolean whether event passes a single-electron trigger
trigM	bool	boolean whether event passes a single-muon trigger
trigP	bool	boolean whether event passes a diphoton trigger
lep_n	int	number of pre-selected leptons
lep_truthMatched	vector<bool>	boolean indicating whether the lepton is matched to a simulated lepton
lep_trigMatched	vector<bool>	boolean indicating whether the lepton is the one triggering the event
lep_pt	vector<float>	transverse momentum of the lepton
lep_eta	vector<float>	pseudo-rapidity, η , of the lepton
lep_phi	vector<float>	azimuthal angle, ϕ , of the lepton
lep_E	vector<float>	energy of the lepton
lep_z0	vector<float>	z -coordinate of the track associated to the lepton wrt. primary vertex
lep_charge	vector<int>	charge of the lepton
lep_type	vector<int>	number signifying the lepton type (e or μ)
lep_isTightID	vector<bool>	boolean indicating whether lepton satisfies tight ID reconstruction criteria
lep_ptcone30	vector<float>	scalar sum of track p_T in a cone of $R=0.3$ around lepton, used for tracking isolation
lep_etcone20	vector<float>	scalar sum of track E_T in a cone of $R=0.2$ around lepton, used for calorimeter isolation
lep_trackd0pvunbiased	vector<float>	d_0 of track associated to lepton at point of closest approach (p.c.a.)
lep_tracksigd0pvunbiased	vector<float>	d_0 significance of the track associated to lepton at the p.c.a.
met_et	float	transverse energy of the missing momentum vector
met_phi	float	azimuthal angle of the missing momentum vector
jet_n	int	number of pre-selected jets
jet_pt	vector<float>	transverse momentum of the jet
jet_eta	vector<float>	pseudo-rapidity, η , of the jet
jet_phi	vector<float>	azimuthal angle, ϕ , of the jet
jet_E	vector<float>	energy of the jet
jet_jvt	vector<float>	jet vertex tagger discriminant [21] of the jet
jet_trueflav	vector<int>	flavour of the simulated jet
jet_truthMatched	vector<bool>	boolean indicating whether the jet is matched to a simulated jet
jet_MV2c10	vector<float>	output from the multivariate b -tagging algorithm [22] of the jet

Table 4: Description of the 13 TeV ATLAS Open Data branches and variables.

Tuple branch name	C++ type	Variable description
photon_n	int	number of pre-selected photons
photon_truthMatched	vector<bool>	boolean indicating whether the photon is matched to a simulated photon
photon_trigMatched	vector<bool>	boolean indicating whether the photon is the one triggering the event
photon_pt	vector<float>	transverse momentum of the photon
photon_eta	vector<float>	pseudo-rapidity of the photon
photon_phi	vector<float>	azimuthal angle of the photon
photon_E	vector<float>	energy of the photon
photon_isTightID	vector<bool>	boolean indicating whether photon satisfies tight identification reconstruction criteria
photon_ptcone30	vector<float>	scalar sum of track p_T in a cone of $R=0.3$ around photon
photon_etcone20	vector<float>	scalar sum of track E_T in a cone of $R=0.2$ around photon
photon_convType	vector<int>	information whether and where the photon was converted
largeRjet_n	int	number of pre-selected large- R jets
largeRjet_pt	vector<float>	transverse momentum of the large- R jet
largeRjet_eta	vector<float>	pseudo-rapidity of the large- R jet
largeRjet_phi	vector<float>	azimuthal angle of the large- R jet
largeRjet_E	vector<float>	energy of the large- R jet
largeRjet_m	vector<float>	invariant mass of the large- R jet
largeRjet_truthMatched	vector<int>	information whether the large- R jet is matched to a simulated large- R jet
largeRjet_D2	vector<float>	weight from algorithm [57] for W/Z -boson tagging
largeRjet_tau32	vector<float>	weight from algorithm [57] for top-quark tagging
tau_n	int	number of pre-selected hadronically decaying τ -lepton
tau_pt	vector<float>	transverse momentum of the hadronically decaying τ -lepton
tau_eta	vector<float>	pseudo-rapidity of the hadronically decaying τ -lepton
tau_phi	vector<float>	azimuthal angle of the hadronically decaying τ -lepton
tau_E	vector<float>	energy of the hadronically decaying τ -lepton
tau_charge	vector<int>	charge of the hadronically decaying τ -lepton
tau_isTightID	vector<bool>	boolean indicating whether hadronically decaying τ -lepton satisfies tight ID reconstruction criteria
tau_truthMatched	vector<bool>	boolean indicating whether the hadronically decaying τ -lepton is matched to a simulated τ -lepton
tau_trigMatched	vector<bool>	boolean signifying whether the τ -lepton is the one triggering the event
tau_nTracks	vector<int>	number of tracks in the hadronically decaying τ -lepton decay
tau_BDTID	vector<float>	output of the multivariate algorithm [24] discriminating hadronically decaying τ -leptons from jets
dtau_m	float	$\Delta\tau$ invariant mass using the missing-mass calculator [54]
lep_pt_syst	vector<float>	single component syst. uncert. (lepton momentum scale and resolution [15, 36]) affecting lep_pt
met_et_syst	float	single component syst. uncert. (E_T^{miss} scale and resolution [30]) affecting met_pt
jet_pt_syst	vector<float>	single component syst. uncert. (jet energy scale [37]) affecting jet_pt
photon_pt_syst	vector<float>	single component syst. uncert. (photon energy scale and resolution [16]) affecting photon_pt
largeRjet_pt_syst	vector<float>	single component syst. uncert. (large- R jet energy resolution [37]) affecting largeRjet_pt
tau_pt_syst	vector<float>	single component syst. uncert. (τ -lepton reconstruction and energy scale [24]) affecting tau_pt

Fuente: Pag. (35-36)

<https://cds.cern.ch/record/2707171/files/ANA-OTRC-2019-01-PUB-updated.pdf>

dict samples

```
In [7]: samples = {  
    'data': {  
        'list' : ['data_A', 'data_B', 'data_C', 'data_D'],  
    },  
  
    r'Background $Z,t\bar{t}$' : { # Z + ttbar  
        'list' : ['Zee', 'Zmumu', 'ttbar_lep'],  
        'color' : "#6b59d3" # purple  
    },  
  
    r'Background $ZZ^*$' : { # ZZ  
        'list' : ['llll'],  
        'color' : "#ff0000" # red  
    },  
  
    r'Signal ($m_H$ = 125 GeV)' : { # H -> ZZ -> llll  
        'list' : ['ggH125_ZZ4lep', 'VBFH125_ZZ4lep', 'WH125_ZZ4lep', 'ZH125_ZZ4lep'],  
        'color' : "#00cdff" # light blue  
    },  
}
```

- 'data': Pertenece a los datos obtenido de algún experimento
- r'Background \$Z,t\bar{t}\$': Pertenece a la simulación de Monte Carlo
- r'Background \$ZZ^*\$': Pertenece a la simulación de Monte Carlo
- r'Signal (\$m_H\$ = 125 GeV)': Pertenece a la simulación de Monte Carlo

Se extraen los datos usando `get_data_from_files()` y `read_file()` y se obtiene un dict con 4 dataframe respectivos para cada key de samples

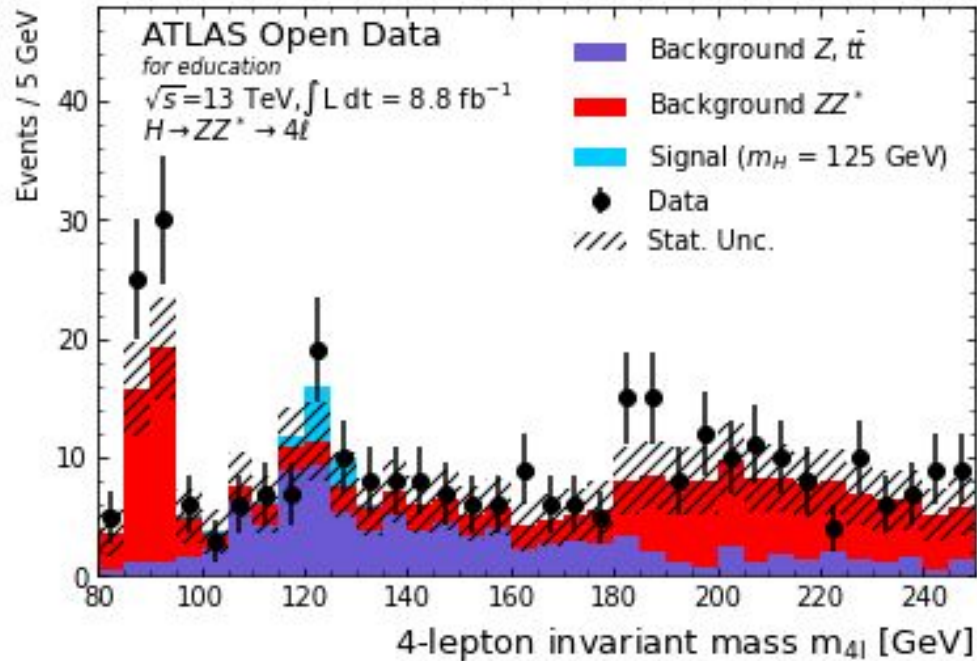
	lep_pt	lep_eta	lep_phi	lep_E	lep_charge	lep_type	mcWeight	scaleFactor_PILEUP	scaleFactor_ELE	scaleFactor_MUON	scaleFactor
entry											
1	[175558.73, 138846.67, 106696.53, 79776.73]	[2.076854, 0.9641987, 0.4538123, 2.3985784]	[1.2929627, -2.2561386, -1.7970102, 0.5870669]	[711421.06, 208545.92, 117873.27, 442695.56]	[-1, 1, -1, 1]	[11, 13, 13, 11]	0.0	0.0	0.0	0.0	
2	[65284.066, 37290.766, 12426.926, 8804.564]	[0.103151694, 0.43131045, 0.98646265, 1.1151001]	[2.3295352, -2.4817212, 2.2234292, 2.4967396]	[65631.695, 40813.45, 18980.092, 14870.211]	[-1, 1, -1, 1]	[11, 11, 13, 13]	0.0	0.0	0.0	0.0	
4	[146784.7, 127539.97, 33562.18, 29776.129]	[-0.20137234, -0.26222968, -0.2997417, 0.47760...	[0.56058794, -2.2311037, 2.0015476, 2.9714534]	[149770.89, 131950.27, 35081.2, 33237.22]	[1, 1, -1, -1]	[11, 11, 11, 11]	0.0	0.0	0.0	0.0	
5	[135680.69, 82894.8, 62508.098, 42552.906]	[1.4913375, 0.73802865, 1.2436671, 0.5755266]	[-0.7563026, -2.7576957, 2.2719328, -1.1226101]	[316685.9, 106514.14, 117410.016, 49797.125]	[-1, -1, 1, 1]	[13, 11, 11, 13]	0.0	0.0	0.0	0.0	
7	[67713.18, 42791.816, 12742.542, 10291.093]	[-1.5264179, -0.06150811, -0.6896634, -0.6163658]	[-1.4037814, -1.353487, -0.32475498, -0.3771284]	[163154.1, 42872.79, 15895.331, 12309.051]	[1, -1, -1, 1]	[11, 11, 13, 13]	0.0	0.0	0.0	0.0	

Previamente se han definido funciones como:

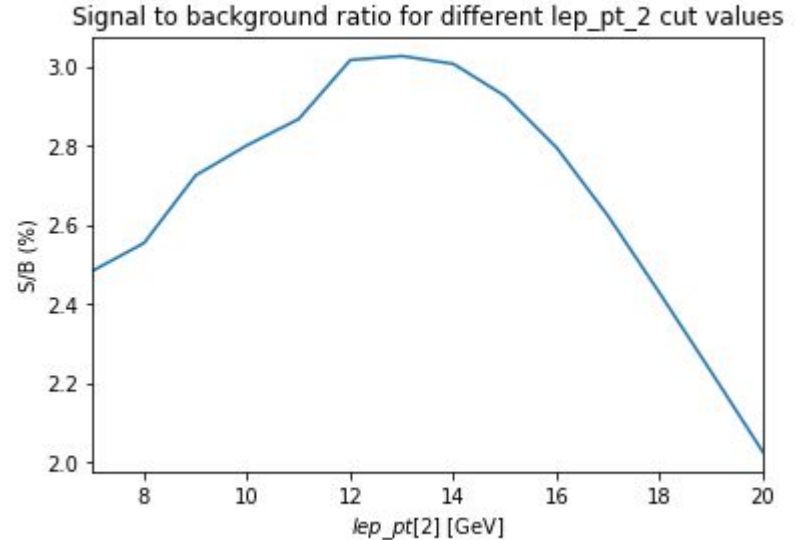
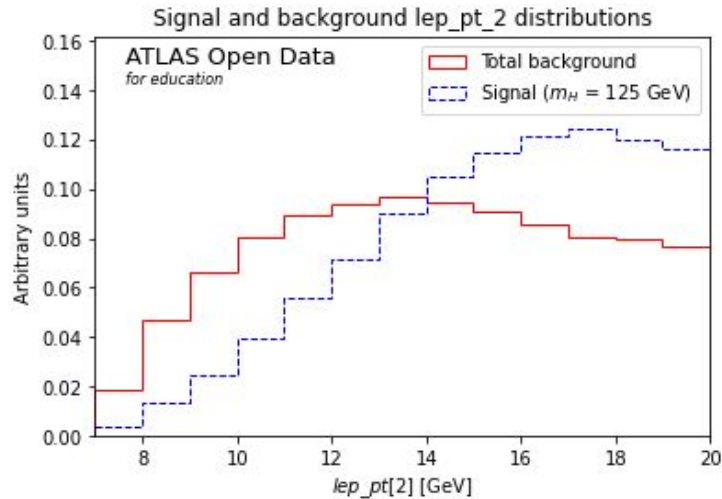
- `calc_weight()`
- `get_xsec_weight()`
- `calc_mllll()`
- `cut_lep_charge()`
- `cut_lep_type()`

que sirven para obtener cantidades físicas, también podrían haber sido después de obtener la data pero por cuestiones de optimización se hace de esta manera

histogramas de data, y MC (backgrounds and signal)



De MC se toma signal and background para compararlos en histograma en por cada lepton



Se obtiene esto para los 4 leptones, este es ejemplo para el lepton 3 (2 según python)

Preparamos la data para entrenar

Se almacena MC en X incluyendo signal and background

```
all_MC = [] # define empty list that will contain all features for the MC
for key in data: # loop over the different keys in the dictionary of dataframes
    if key!='data': # only MC should pass this
        all_MC.append(data_for_BDT[key]) # append the MC dataframe to the list containing all MC features
X = np.concatenate(all_MC) # concatenate the list of MC dataframes into a single 2D array of features, called X
```

Se obtiene los shape de signal and background para etiquetar los eventos con 0 y 1 según corresponda

```
all_y = [] # define empty list that will contain labels whether an event is signal or background
for key in data: # loop over the different keys in the dictionary of dataframes
    if key!=r'Signal ($m_H$ = 125 GeV)' and key!='data': # only background MC should pass this
        all_y.append(np.zeros(data_for_BDT[key].shape[0])) # background events are labelled with 0
all_y.append(np.ones(data_for_BDT[r'Signal ($m_H$ = 125 GeV)'].shape[0])) # signal events are labelled with 1
y = np.concatenate(all_y) # concatenate the list of labels into a single 1D array of labels, called y
```

Buscamos el mejor max_depth para AdaBoost

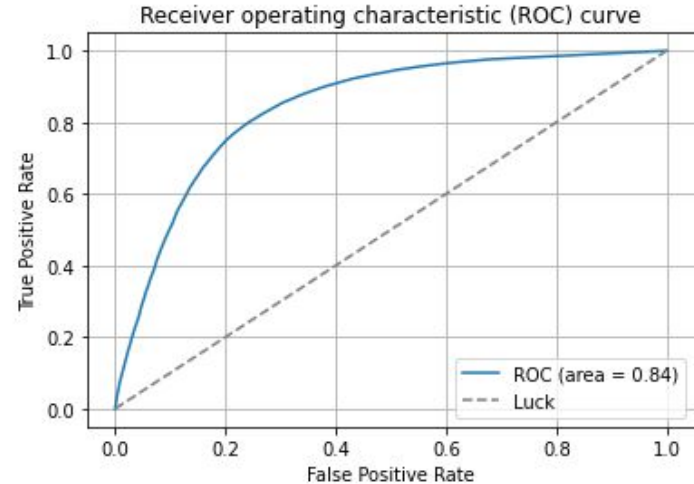
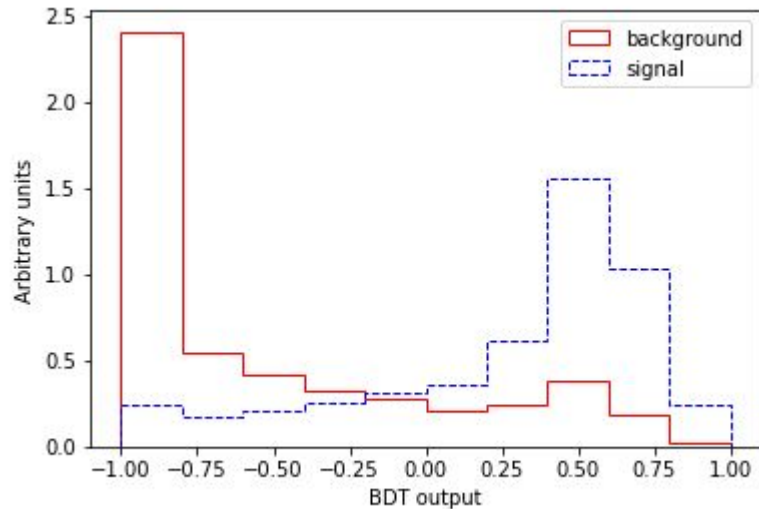
Buscamos el max_depth con mejor accuracy

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score

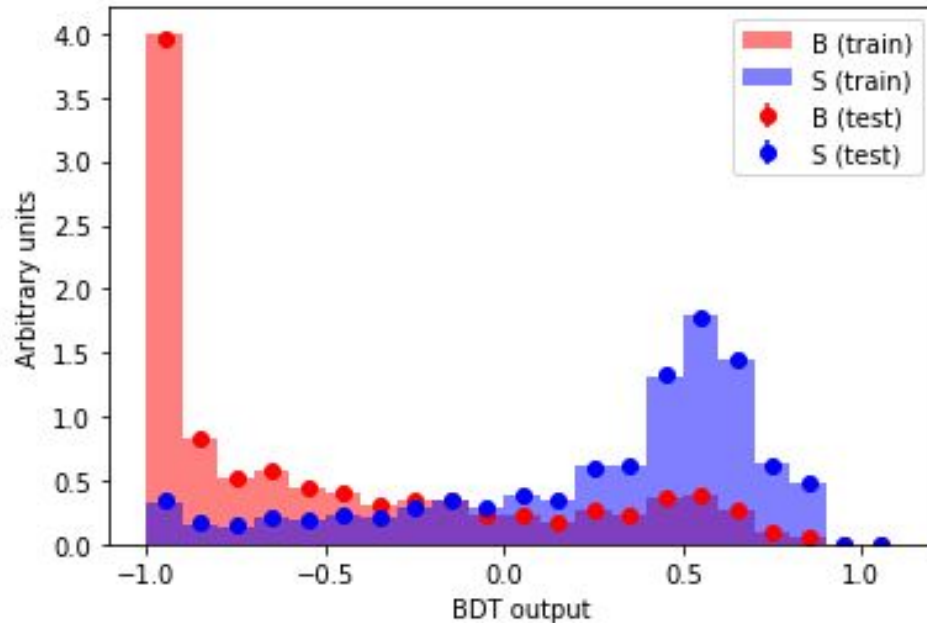
for i in range(1,10):
    dt = DecisionTreeClassifier(max_depth=i) # maximum depth of the tree
    bdt = AdaBoostClassifier(dt,
                            algorithm='SAMME', # SAMME discrete boosting algorithm
                            n_estimators=12, # max number of estimators at which boosting is terminated
                            learning_rate=0.5) # shrinks the contribution of each classifier by learning_rate
    bdt.fit(X_train, y_train)

    y_pred = bdt.predict(X_test)
    print("Time taken to fit BDT: "+str(round(elapsed,1))+ "s") # print total time taken to fit BDT
    print("Mi árbol da un accuracy de:", accuracy_score(y_test,y_pred), "cuando su max_depth es: ", i)
```

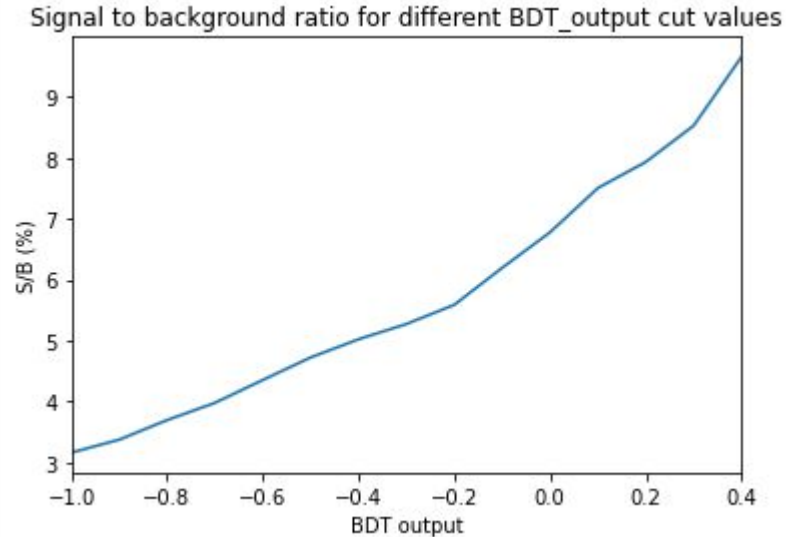
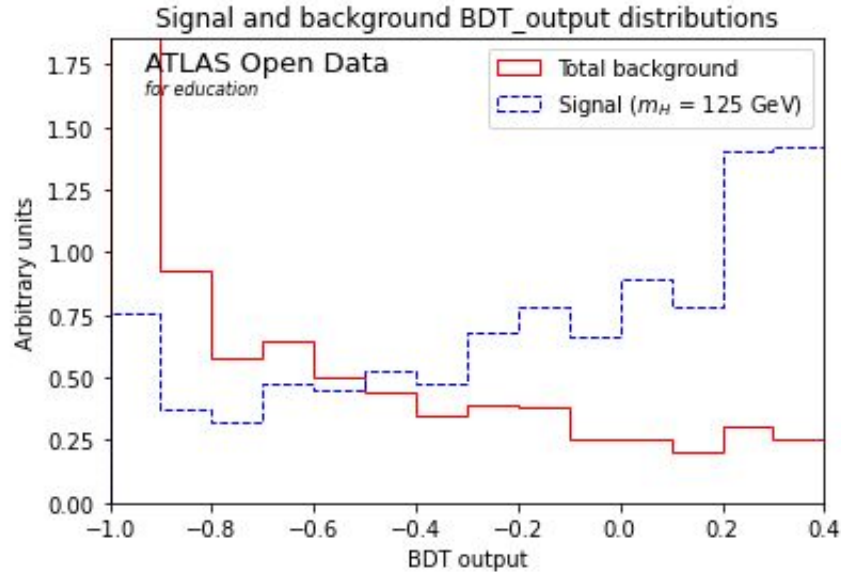

Entrenamos y luego ploteamos para identificar el desempeño, se usa como metrica la curva ROC



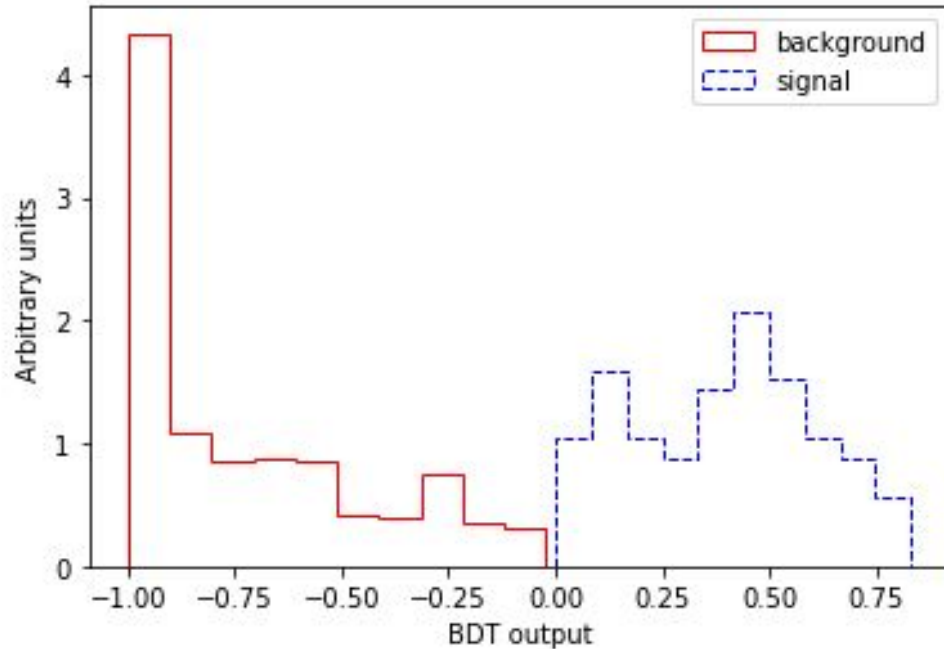
Esto es una comparación de los datos usados para el entrenamiento y testeo



Se plotea BTD outputs y se analiza la relación S/B



Usando el aprendizaje obtenido vuelvo a entrenar pero con los datos experimentales, data['data']



Obtengo esto, que no es lo que esperaba, asumo que me equivoque pero no logré encontrar en que.

Comentarios

- La data fue un tanto compleja entenderla al principio pero luego cuando encontrar el `infofile.py` me permitio entender
- Encontre un notebook en donde hacían un BDT para 2 lep y con solo un fondo y una señal, intenté adaptarlo para 4 leptones y con 2 fondos y una señal
- Al final quise probar los parámetros obtenidos en el entrenamiento para predecir la data experimental pero no obtuve lo que esperaba, no tengo con que comparar así que no puedo presentar una curva ROC
- Todo esto es muy interesante y emocionante.