

# CxAODReader setup on lxplus

```
[schioppa@lxplus799:~/Reader4/CxAODReader_VVSemileptonic] stat -c %y .git/FETCH_HEAD  
2021-07-13 17:57:51.000000001 +0200
```

Code version from 13-07-2021

```
setupATLAS  
asetup AnalysisBase, 21.2.150, here  
source build/x86_64-centos7-gcc8-opt/setup.sh
```

Environment settings. In particular, no local installation (e.g. lwttn) → **compile**

```
# N events  
int maxEvents = -1 # -1 = all events  
  
#Apply pile-up reweighting  
bool applyPUWeight = true  
vector<string> ilumicalcFiles = GoodRunsLists/data15_13TeV/20170619/PHYS_StandardGRL_All_Good_25ns_276262-284484_1_297730-311481_ofLLumi-13TeV-009.root  
vector<string> configFiles = $WorkDir_DIR/data/CxAODOperations_VHbb/PRW/PRW.VHbb.mc16a.root  
  
# input directory where the CxAODs are stored  
# HIGGS5D1 = 0 leptons, HIGGS5D2 = 1 lepton, HIGGS2D4 = 2 leptons  
#string dataset_dir = /eos/user/m/mmazza/samples/Wh_signal_samples/MC16a/  
string dataset_dir = /eos/user/r/rles/CxAOD/r33-08/TCC_EMTopo/HIGGS2D4/  
  
# List of samples to run on:  
#vector<string> samples = group.phys-exotics.mc16_13TeV.302341.CAOD_HIGGS5D2.e7767_e5984_s3126_r9364_r9315_p3641.3:_s3126_r9364_r9315_p3641_22-16-TTADIFTX-19_CxAOD.root  
vector<string> samples = Diboson Signal
```

Full statistics on signal and diboson

**Not going to touch this config file anymore**

```
[schioppa@lxplus799:~/Reader4/run] ls -lrth output_2021-09-08_test/fetch/data-MVATree/  
total 433M  
-rw-r--r--. 1 schioppa zp 65K 8 set 22.51 Diboson-2.root  
-rw-r--r--. 1 schioppa zp 21M 8 set 22.54 Signal-0.root  
-rw-r--r--. 1 schioppa zp 147M 8 set 23.12 Diboson-1.root  
-rw-r--r--. 1 schioppa zp 266M 8 set 23.23 Diboson-0.root
```

Output

# Building the DNN

```
In [2]: inputFiles = ['Signal-0','Diboson-0','Diboson-1','Diboson-2']
for i in inputFiles:
    inFile = '../ntuples/original/'+i+'.root'
    theFile = uproot3.open(inFile)
    tree = uproot3.open(inFile)['Nominal']
    Nevents = uproot3.open(inFile)['Nominal'].numentries
    print ('Number of events in '+inFile,'\t'+str(Nevents))
```

```
Number of events in ../ntuples/original/Signal-0.root 67836
Number of events in ../ntuples/original/Diboson-0.root 1184019
Number of events in ../ntuples/original/Diboson-1.root 621719
Number of events in ../ntuples/original/Diboson-2.root 100
```

Number of events

```
In [3]: DF_Signal = uproot3.open('../ntuples/original/Signal-0.root')['Nominal'].pandas.df()
```

```
In [4]: DF_Diboson = uproot3.open('../ntuples/original/Diboson-1.root')['Nominal'].pandas.df()
DF_Diboson.shape
```

```
Out[4]: (621719, 225)
```

I consider just one background file (Diboson-1.root)

```
In [5]: import sklearn.utils
def Shuffling(df):
    df = sklearn.utils.shuffle(df,random_state=123)
    df = df.reset_index(drop=True)
    return df
```

```
DF_Signal = Shuffling(DF_Signal)
DF_Diboson = Shuffling(DF_Diboson)
```

```
In [6]: DF_Signal.insert(len(DF_Signal.columns), 'isSignal', np.ones(DF_Signal.shape[0]), True)
DF_Diboson.insert(len(DF_Diboson.columns), 'isSignal', np.zeros(DF_Diboson.shape[0]), True)
```

```
signalList = [DF_Signal[:50000] ]
dibosonList = [DF_Diboson[:50000] ]
```

I take 50k sig + 50k bkg

```
totalPD_sig = pd.concat(signalList,ignore_index=True)
totalPD_bkg = pd.concat(dibosonList,ignore_index=True)
```

```
print ('Signal events:',totalPD_sig.shape)
print ('Background events:',totalPD_bkg.shape)
```

```
df_total = pd.concat([totalPD_sig,totalPD_bkg],ignore_index=True)
df_total = Shuffling(df_total)
print ("Total events after shuffling: ",df_total.shape)
```

# Building the DNN

```
In [7]: InputFeatures = ['lep1_m', 'lep1_pt', 'lep1_eta', 'lep1_phi']# ← 2_m', 'lep2_pt', 'lep2_eta', 'lep2_phi', 'fatjet'
X=df_total[InputFeatures].values
y_tmp = df_total['isSignal']
eventID = df_total['eventNumber']
w = np.ones(X.shape[0])
```

```
In [8]: from sklearn.preprocessing import StandardScaler,LabelEncoder
scaler = StandardScaler()
X = scaler.fit_transform(X)
le = LabelEncoder()
y = le.fit_transform(y_tmp)
```

```
In [9]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test, w_train, w_test = train_test_split(X, y, w, train_size=0.7,random_state=123)
```

```
In [10]: from keras.models import Model, Sequential
from keras.layers import Dense, Dropout, Input, BatchNormalization
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.layers.core import Dense, Activation

def BuildDNN(N_input,width,depth):
    model = Sequential()
    model.add(Dense(units=width, input_dim=N_input))
    model.add(Activation('relu'))
    model.add(Dropout(0.2))
    # First layer
    for i in range(0, depth):
        model.add(Dense(width))
        model.add(Activation('relu'))
        # Dropout randomly sets a fraction of input units to 0 at each update during training time
        # which helps prevent overfitting.
        model.add(Dropout(0.2))
        model.add(Dense(1, activation='sigmoid'))
    # Output layer/node
    return model

n_dim=X_train.shape[1]
n_nodes = 32
n_depth = 2 # number of additional hidden layers
```

```
In [11]: model=BuildDNN(n_dim,n_nodes,n_depth)
model.compile(loss='binary_crossentropy',optimizer='rmsprop',metrics=['accuracy'])
```

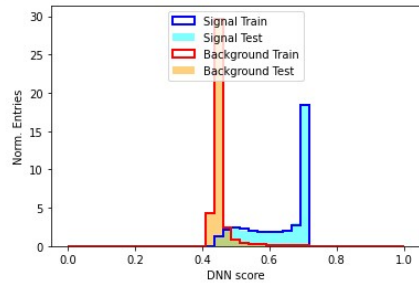
```
In [12]: modelMetricsHistory = model.fit(X_train, y_train, epochs=4,batch_size=2048,validation_split=0.2, verbose=1)
```

Purposely downgrading the DNN (I want to have a large variation of scores)

# Building the DNN

## Saving the model and architecture

```
plt.hist(yhat_train_signal, bins=bins, histtype='step', lw=2, color='blue', label=[r'Signal Train'], density=True)
plt.hist(yhat_test_signal, bins=bins, histtype='stepfilled', lw=2, color='cyan', alpha=0.5, label=[r'Signal Test'], de
plt.hist(yhat_train_background, bins=bins, histtype='step', lw=2, color='red', label=[r'Background Train'], density=Tr
plt.hist(yhat_test_background, bins=bins, histtype='stepfilled', lw=2, color='orange', alpha=0.5, label=[r'Background
plt.ylabel('Norm. Entries')
plt.xlabel('DNN score')
plt.savefig("output/scores.pdf")
plt.show()
plt.clf()
```



<Figure size 432x288 with 0 Axes>

```
In [20]: def SaveArchAndWeights(model, outputDir):
arch = model.to_json()
outputArch = outputDir + '/architecture.json'
with open(outputArch, 'w') as arch_file:
    arch_file.write(arch)
outputWeights = outputDir + '/weights.h5'
model.save_weights(outputWeights)

SaveArchAndWeights(model, 'output')
```

# Building the DNN

```
X=df_total[InputFeatures].values
y_tmp = df_total['isSignal']
eventID = df_total['eventNumber']
w = np.ones(X.shape[0])
```

```
In [8]: with open('output/variables.json', 'w') as var_file:
        var_file.write("{\n")
        var_file.write("  \"inputs\": [\n")
        index = 0
        for col in X.transpose():
            offset = -1. * np.mean(col)
            scale = 1. / np.std(col)
            print(-offset)
            var_file.write("    {\n")
            var_file.write("      \"name\": \"%s\", \n" % InputFeatures[index])
            var_file.write("      \"offset\": %lf, \n" % offset)
            var_file.write("      \"scale\": %lf \n" % scale)
            var_file.write("    } \n")
            if index != (len(InputFeatures)-1):
                var_file.write(", \n")
            else:
                var_file.write("\n")
            index = index + 1
        var_file.write("  ], \n")
        var_file.write("  \"class_labels\": [\"BinaryClassificationOutputName\"] \n")
        var_file.write("} \n")
```

```
0.0547628420e-02
336.2005615234375
0.0004946031258441508
0.01665002666413784
```

```
In [9]: from sklearn.preprocessing import StandardScaler, LabelEncoder
        scaler = StandardScaler()
        X = scaler.fit_transform(X)
        le = LabelEncoder()
        y = le.fit_transform(y_tmp)
        print(scaler.mean_)
```

```
[5.47628420e-02 3.36200609e+02 4.94603151e-04 1.66500305e-02]
```

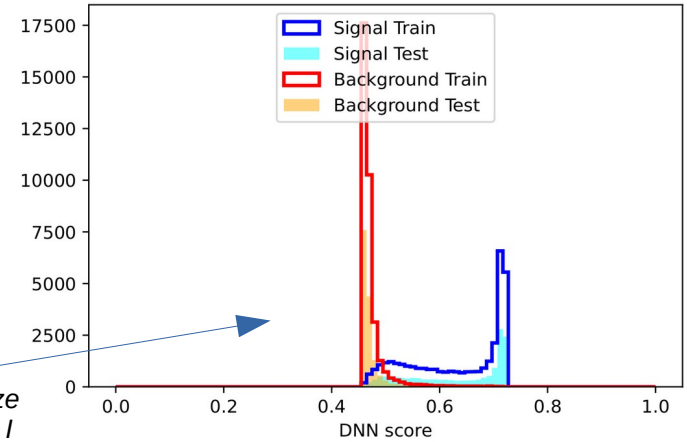
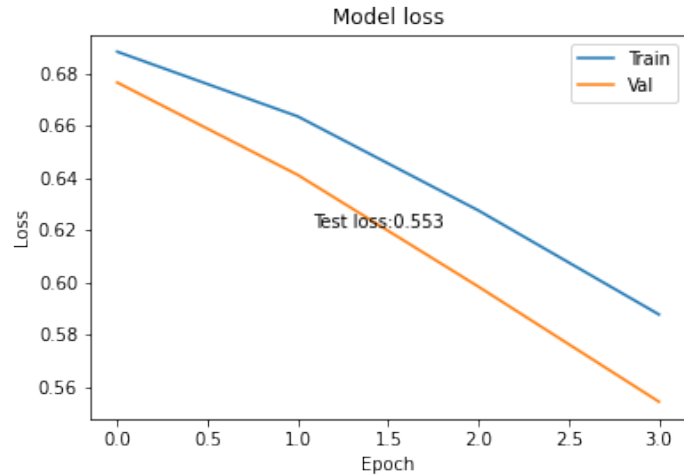
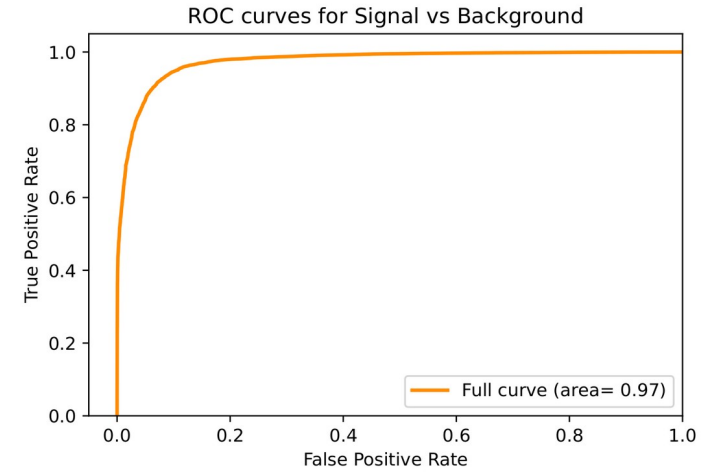
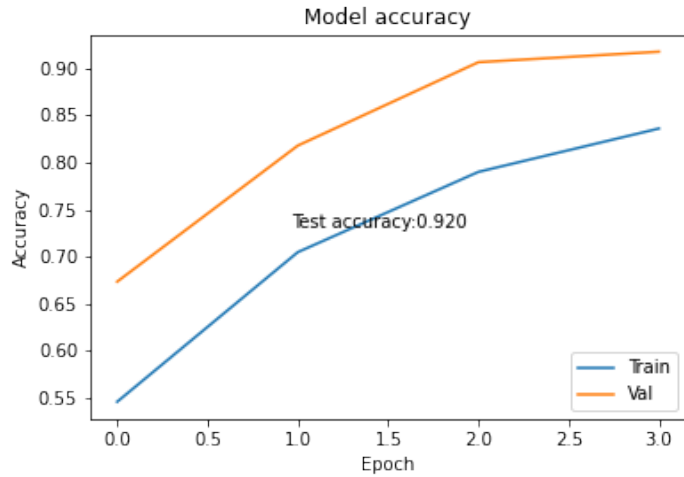
```
In [ ]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test, w_train, w_test = train_test_split(X, y, w, train_size=0.7, random_state=123)
```

```
In [ ]: from keras.models import Model, Sequential
        from keras.layers import Dense, Dropout, Input, BatchNormalization
```

Saving the variables file

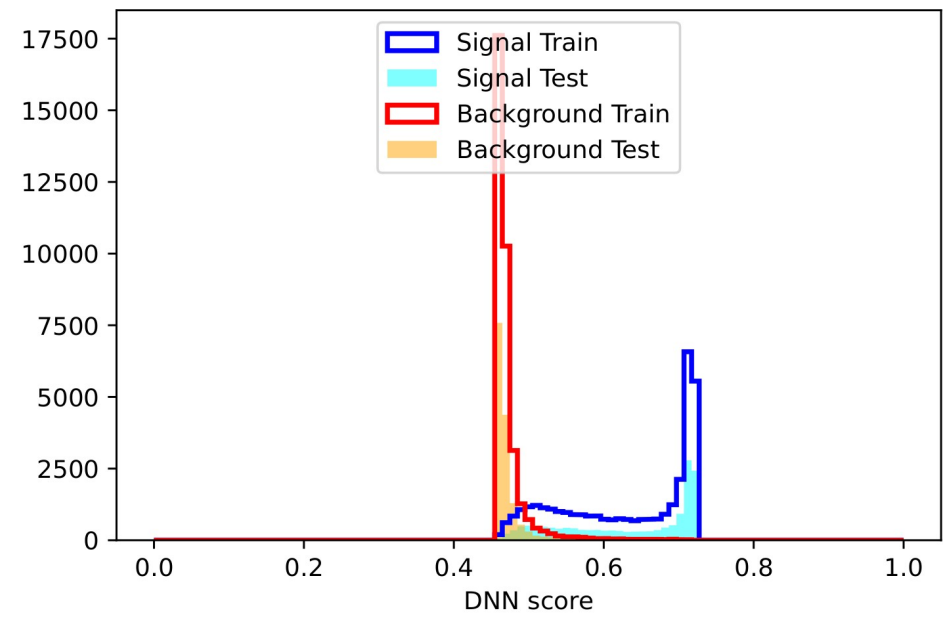
Sanity-checked that the mean and std calculated by the StandardScaler are the same that are saved in the variables file

# Results

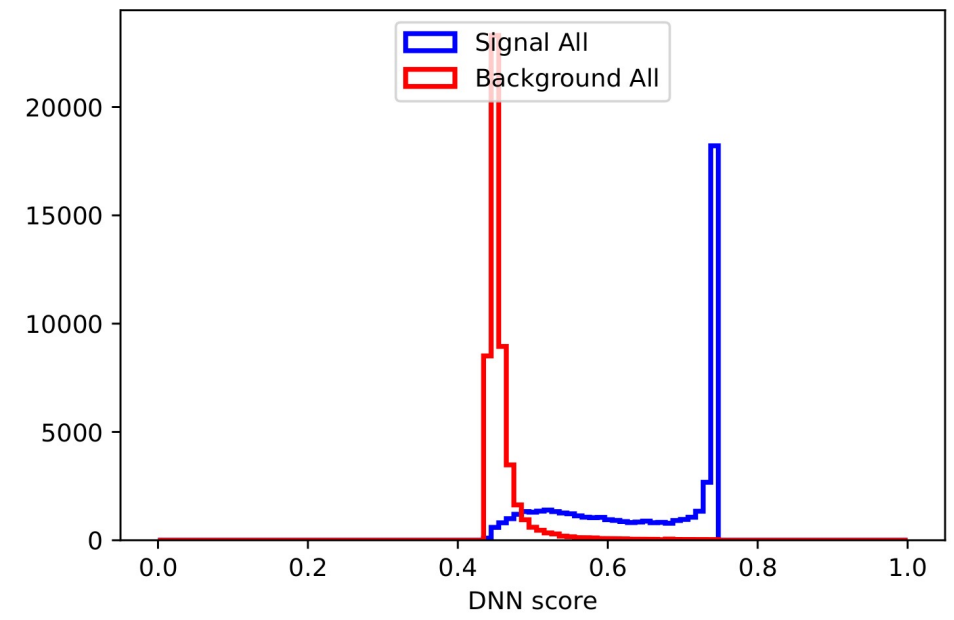


*I don't want to normalize the score distributions, I want to see the absolute scales*

# Results



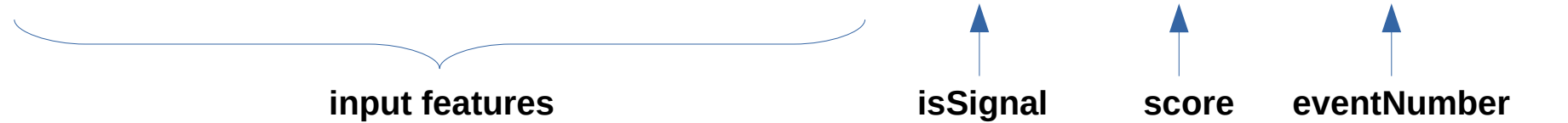
## Full set





# Keeping eventNumber

```
1.056583747267723083e-01 7.5515144534814453125e+01 -9.953287859889520307e-01 -5.012756109257670898e+00 0.0000000000000000e+00 3.721019527040533447e-01 4.5844520000000000e+00
5.109951598569750786e-04 7.492948150634765625e+01 1.375156998634338379e+00 2.38028740882835352e+00 0.0000000000000000e+00 3.645699620246887207e-01 1.7458330000000000e+06
1.056583747267723083e-01 6.444206542968750000e+02 -7.39238619804382342e-01 1.029330492019653320e+00 1.0000000000000000e+00 5.502515435218811035e-01 3.0038000000000000e+04
1.056583747267723083e-01 3.961435241699218750e+02 -1.120612978935241699e+00 -1.356585741043090820e+00 1.0000000000000000e+00 5.317182540893554688e-01 3.3607000000000000e+04
1.056583747267723083e-01 9.946021118164062500e+02 -1.053907155990600586e+00 3.058271884918212891e+00 1.0000000000000000e+00 5.516743063926696777e-01 4.7660000000000000e+06
5.109988851472735405e-04 5.312201309204101562e+01 1.013874888420104980e+00 -2.262141704559326172e+00 0.0000000000000000e+00 3.402373790740966797e-01 3.8858970000000000e+06
1.056583747267723083e-01 1.521951416015625000e+03 -6.534660607576370239e-02 1.317083477973937988e+00 1.0000000000000000e+00 5.516642332077026367e-01 7.1020000000000000e+03
1.056583747267723083e-01 6.490019226074218750e+01 -6.3218319416006414258e-01 2.355821847915649414e+00 0.0000000000000000e+00 3.578881621360778809e-01 2.7722070000000000e+06
1.056583747267723083e-01 4.825260314941406250e+02 1.696055889129638672e+00 -2.241934061050415039e+00 1.0000000000000000e+00 5.367501974105834961e-01 2.1354000000000000e+04
1.056583747267723083e-01 6.847990112304687500e+02 -4.610644280910491943e-01 -1.831165313720703125e+00 1.0000000000000000e+00 5.517069101333618164e-01 2.5785000000000000e+04
5.109988851472735405e-04 5.000741577148437500e+01 1.279896616935729980e+00 -2.886009931564331055e+00 0.0000000000000000e+00 3.384260535240173340e-01 3.9492610000000000e+06
5.109977209940552711e-04 3.939311981201171875e+01 5.003802180290222168e-01 1.903962373733520508e+00 0.0000000000000000e+00 3.482488393783569336e-01 2.4592680000000000e+06
5.113458610139787197e-04 7.885030517578125000e+02 -1.071359276771545410e+00 3.084963083267211914e+00 1.0000000000000000e+00 5.517007708549499512e-01 3.1896000000000000e+04
1.056583672761917114e-01 7.490515136718750000e+02 -1.890387177467346191e+00 -1.177463531494140625e+00 1.0000000000000000e+00 5.515177845954895020e-01 3.7627000000000000e+04
5.109988851472735405e-04 5.825258255004882812e+01 5.568439960479736328e-01 1.557882428169250488e+00 0.0000000000000000e+00 3.50704997779388428e-01 8.6115700000000000e+05
5.109988851472735405e-04 4.635451126098632812e+01 -1.44252039909362793e+00 7.423601150512695312e-01 0.0000000000000000e+00 3.450210690498352051e-01 9.7446500000000000e+05
1.056583672761917114e-01 8.377225952148437500e+02 1.524820327758789062e+00 2.506663322448730469e+00 1.0000000000000000e+00 5.516879558563232422e-01 2.5841000000000000e+04
5.108682089485228062e-04 4.932612915039062500e+02 1.608679771423339844e+00 -2.924324274066110352e+00 0.0000000000000000e+00 5.413503646850585938e-01 9.5350000000000000e+05
1.056583747267723083e-01 6.040439605712890625e+01 -4.364538788795471191e-01 -1.273103594779968262e+00 0.0000000000000000e+00 3.481045067310333252e-01 5.6621100000000000e+05
5.109988851472735405e-04 1.791098785400390625e+02 1.812725961208343506e-01 1.200174808502197266e+00 1.0000000000000000e+00 4.101917743682861328e-01 2.3595000000000000e+04
1.056583747267723083e-01 3.465023803710937500e+01 -1.208344101905822754e+00 -2.974064111709594727e+00 0.0000000000000000e+00 3.556953072547912598e-01 1.9582970000000000e+06
1.056583747267723083e-01 6.788417053222656250e+01 2.140004396438598633e+00 3.529804199934005737e-02 0.0000000000000000e+00 3.384172618389129639e-01 5.2145200000000000e+06
5.110007477924227715e-04 3.587042236328125000e+01 -1.818514347076416016e+00 1.101696610450744629e+00 0.0000000000000000e+00 3.411310911178588867e-01 3.1565700000000000e+06
1.056583747267723083e-01 9.330418090820312500e+02 -1.720240414142608643e-01 6.915085315704345703e-01 1.0000000000000000e+00 5.516864061355590820e-01 3.2559000000000000e+04
1.056583747267723083e-01 5.084196777343750000e+02 2.96888024288177490e-01 2.830872774124145508e+00 1.0000000000000000e+00 5.488460063934326172e-01 1.8965000000000000e+04
1.056583747267723083e-01 5.051704788208007812e+01 -6.857565641403198242e-01 -1.410142779350280762e+00 0.0000000000000000e+00 3.468869924545288086e-01 7.0408900000000000e+05
5.109970225021243095e-04 6.568009185791015625e+01 -1.273696966943740845e-01 -2.628883361816406250e+00 0.0000000000000000e+00 3.572551012039184570e-01 1.5646600000000000e+06
1.056583747267723083e-01 2.047597198486328125e+02 2.307907104492187500e+00 -2.556920766830444336e+00 0.0000000000000000e+00 3.713893890380859375e-01 3.6716360000000000e+06
5.111070931889116764e-04 1.149583129882812500e+03 8.438234925270080566e-01 4.011866748332977295e-01 1.0000000000000000e+00 5.516678094863891602e-01 9.2340000000000000e+03
5.109970225021243095e-04 6.859098815917968750e+01 -4.022270739078521729e-01 -6.625410914421081543e-01 0.0000000000000000e+00 3.543222546577453613e-01 3.3996640000000000e+06
5.109970225021243095e-04 6.775487518310546875e+01 5.787957925349473953e-03 1.718319416064142578e+00 0.0000000000000000e+00 3.537316918373107910e-01 5.1396320000000000e+06
1.056583747267723083e-01 6.072145462036132812e+01 -2.033484697341918945e+00 1.300963163375854492e+00 0.0000000000000000e+00 3.519476056098937988e-01 5.2632950000000000e+06
5.109988851472735405e-04 6.405641937255859375e+01 -1.460147500038146973e+00 -2.88035834960937500e+00 0.0000000000000000e+00 3.564865589141845703e-01 1.0196810000000000e+06
```



input features

isSignal

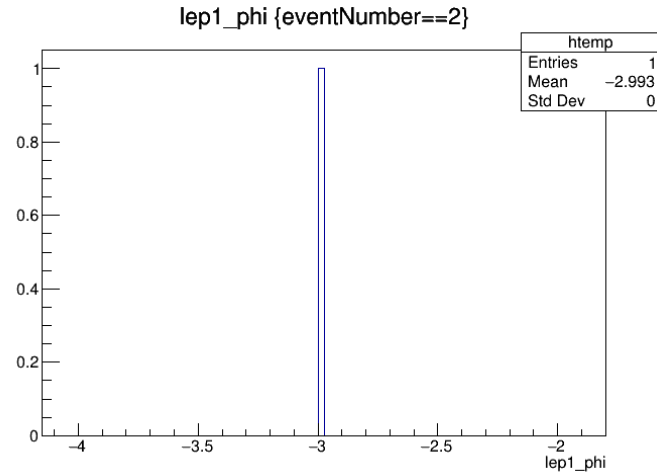
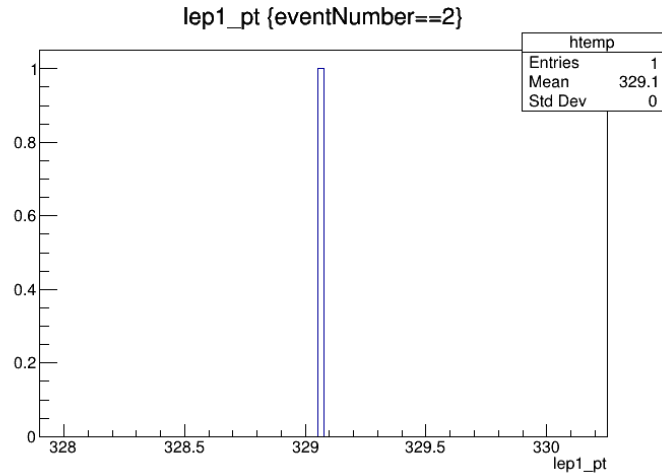
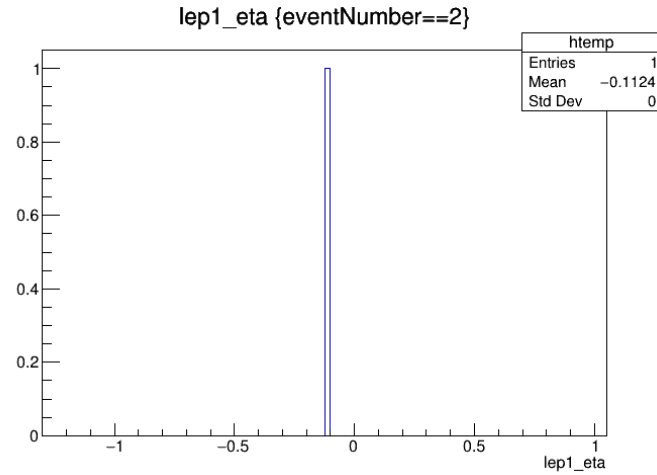
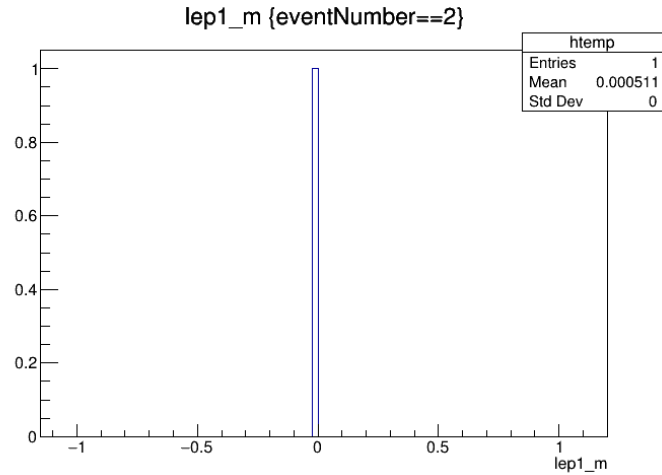
score

eventNumber

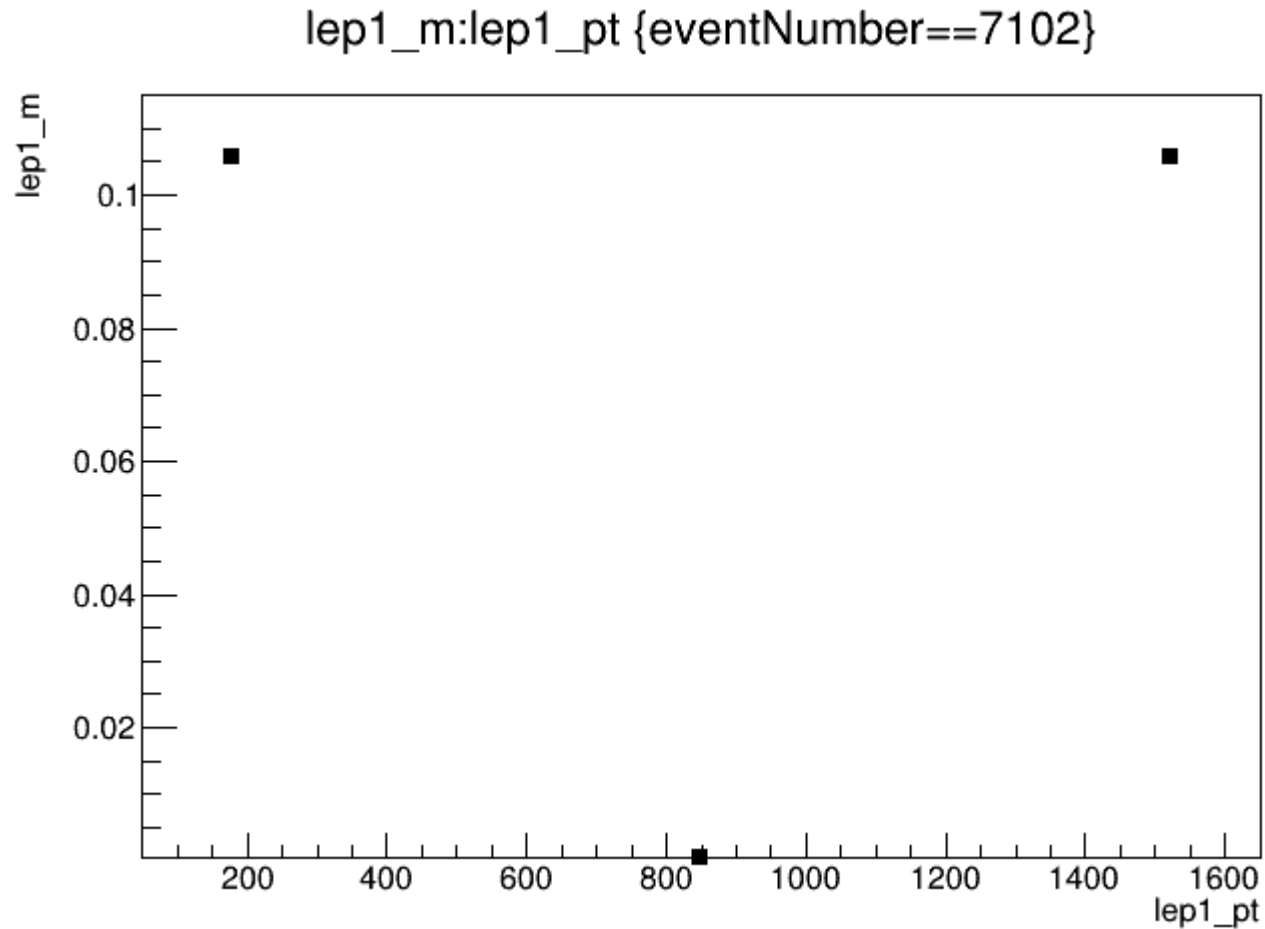


# Keeping eventNumber: sanity check

```
5.109951598569750786e-04 1.675908660888671875e+02 6.767552495002746582e-01 8.211193233728408813e-02 0.0000000000000000e+00 4.007129073143005371e-01 5.097538000000000000e+06  
5.109876510687172413e-04 3.290717773437500000e+02 -1.123887151479721069e-01 -2.993021965026855469e+00 1.0000000000000000e+00 5.03355145444067383e-01 2.0000000000000000e+00  
5.109951598569750786e-04 9.725639343261718750e+01 1.050685882568359375e+00 7.687893509864807129e-01 0.0000000000000000e+00 3.588820099830627441e-01 3.2355220000000000e+06  
1.056583747267723083e-01 2.153034057617187500e+02 1.816065907478332520e+00 -2.044371366500854492e+00 0.0000000000000000e+00 3.866305649280548096e-01 1.4479720000000000e+06
```



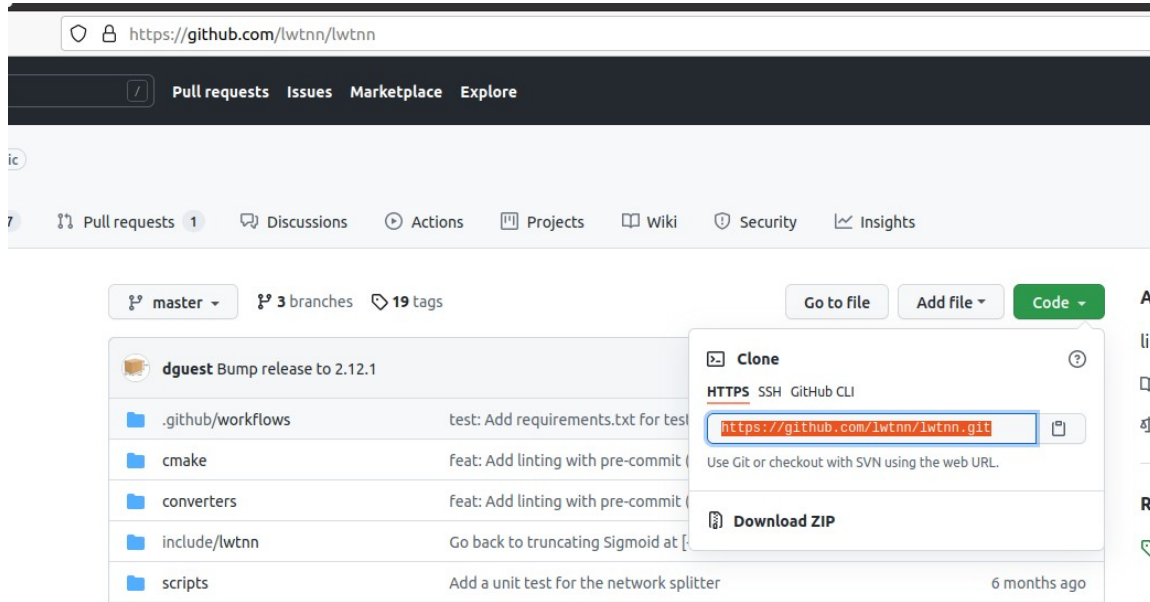
## Keeping eventNumber: observation



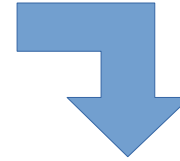
EventNumber is not a unique identifier

From this moment on, I'm not touching python anymore (lest I re-initialize the random generators and produce slightly different results)

# Running the lwttn converters



Cloned a local copy of lwttn



```
enricojr@enricojr-ET2321I:~/cernbox/analysis/development/2021-09-09_lwttnTest$ ll
total 20
drwxrwxr-x  5 enricojr enricojr 4096 set 10 15:58 ./
drwxrwxr-x  9 enricojr enricojr 4096 set  9 10:20 ../
drwxrwxr-x  4 enricojr enricojr 4096 set 10 15:54 AI/
drwxrwxr-x 10 enricojr enricojr 4096 set 10 15:58 lwttn/
drwxrwxr-x  3 enricojr enricojr 4096 set  9 10:20 ntuples/
```

# Running the lwttn converters

```
File Edit Options Buffers Tools Sh-Script Help
# /bin/bash
/home/enricojr/cernbox/analysis/development/2021-09-09_lwttnTest/lwttn/converters/keras2json.py architecture.json variables.json weights.h5 > neural_net.json
```



```
(AI) enricojr@enricojr-ET2321I:~/cernbox/analysis/development/2021-09-09_lwttnTest/AI/output$ ll
total 2620
drwxrwxr-x 2 enricojr enricojr  4096 set 10 16:02 ./
drwxrwxr-x 4 enricojr enricojr  4096 set 10 15:54 ../
-rw-rw-r-- 1 enricojr enricojr 16823 set 10 15:53 Accuracy.png
-rw-rw-r-- 1 enricojr enricojr  3254 set 10 15:53 architecture.json
-rwxrwxr-x 1 enricojr enricojr   170 set 10 16:02 convert*
-rw-rw-r-- 1 enricojr enricojr   142 set 10 16:01 convert~
-rw-rw-r-- 1 enricojr enricojr 17188 set 10 15:53 Loss.png
-rw-rw-r-- 1 enricojr enricojr 40573 set 10 16:02 neural_net.json
-rw-rw-r-- 1 enricojr enricojr 19353 set 10 15:53 ROC.pdf
-rw-rw-r-- 1 enricojr enricojr 13777 set 10 15:53 scores.pdf
-rw-rw-r-- 1 enricojr enricojr 374275 set 10 15:53 scores_test_background.txt
-rw-rw-r-- 1 enricojr enricojr 375725 set 10 15:53 scores_test_signal.txt
-rw-rw-r-- 1 enricojr enricojr 875725 set 10 15:53 scores_train_background.txt
-rw-rw-r-- 1 enricojr enricojr 874275 set 10 15:53 scores_train_signal.txt
-rw-rw-r-- 1 enricojr enricojr   435 set 10 15:53 variables.json
-rw-rw-r-- 1 enricojr enricojr 30960 set 10 15:53 weights.h5
```

## Re-running the CxAODReader

Copied the converted file to the PDNNModel folder:

```
[schioppa@lxplus774:~/Reader4/build/x86_64-centos7-gcc8-opt/data/CxAODReader_VVSemileptonic/PDNNModel] ll
total 42
-rw-r--r--. 1 schioppa zp 1873 Jul 15 09:51 FeatureScaling.dat
-rw-r--r--. 1 schioppa zp 40573 Sep 10 16:06 neural_net.json
```

Notice that this folder already contained a FeatureScaling.dat file too  
According to previous experience, this file is useless (feature scaling is applied by lwttn internally using all the information inside the json file)



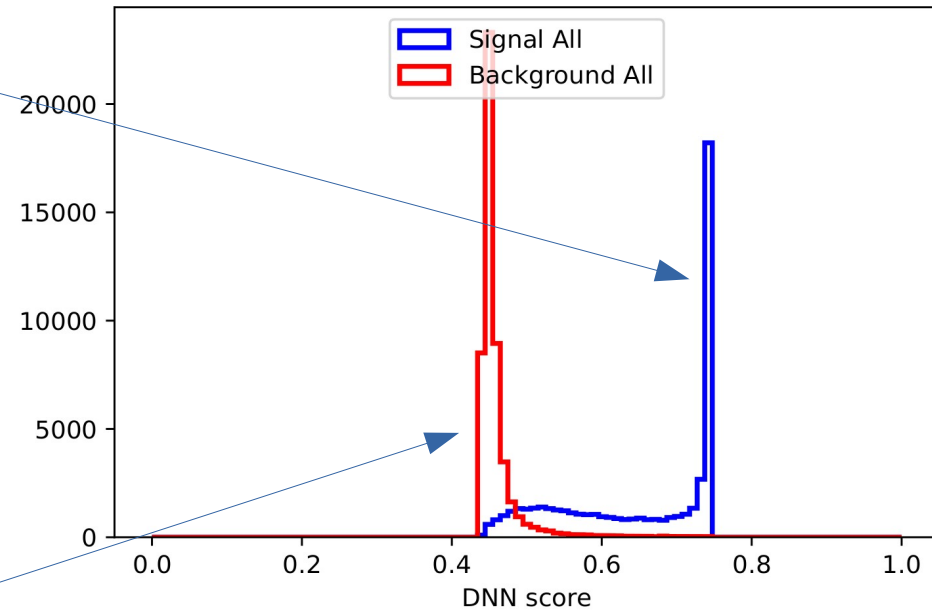
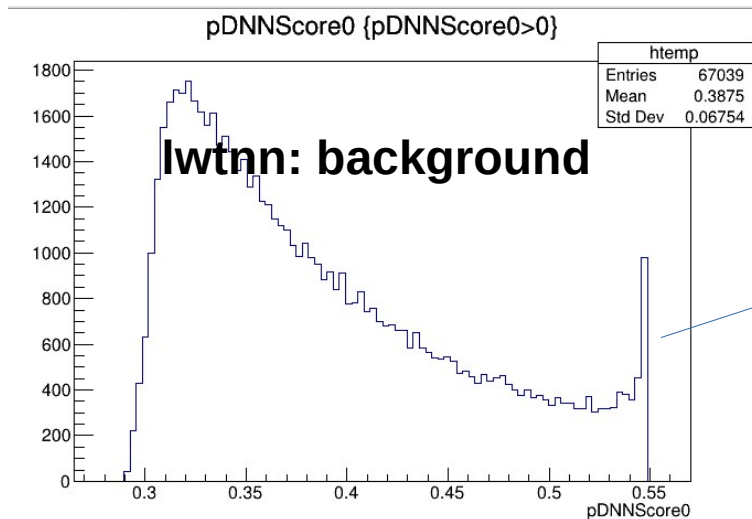
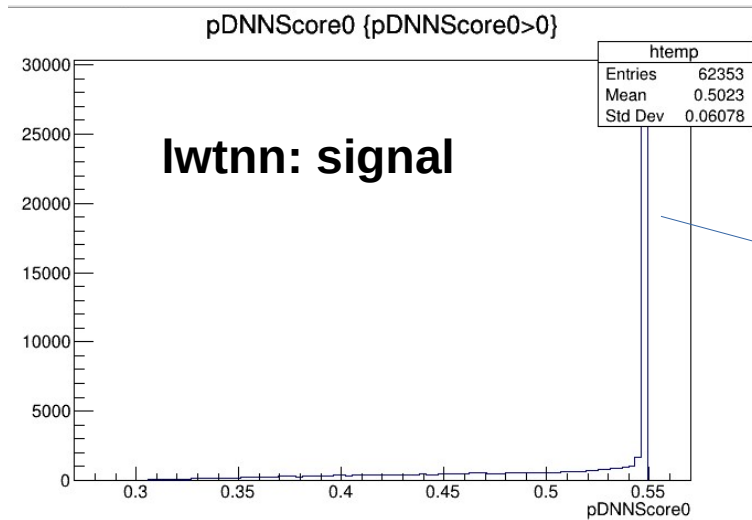
# Re-running the CxAODReader

Configuration file:

```
# PDNN configuration
bool LoadPDNN = true
string PDNNName = pDNN_ggF_Merged
string PDNNModelPath = /data/CxAODReader_VVSemileptonic/PDNNModel/neural_net.json
string PDNNScalingPath = /data/CxAODReader_VVSemileptonic/PDNNModel/FeatureScaling.dat
vector<int> PDNNMassHypothesis = 0 #700 1000 3000 5000
```

One single (dummy) mass hypothesis should collapse the pDNN into a DNN

# Comparing scores



There are arguments why at this level it could be ok that they don't match.  
However ...

# Comparing scores

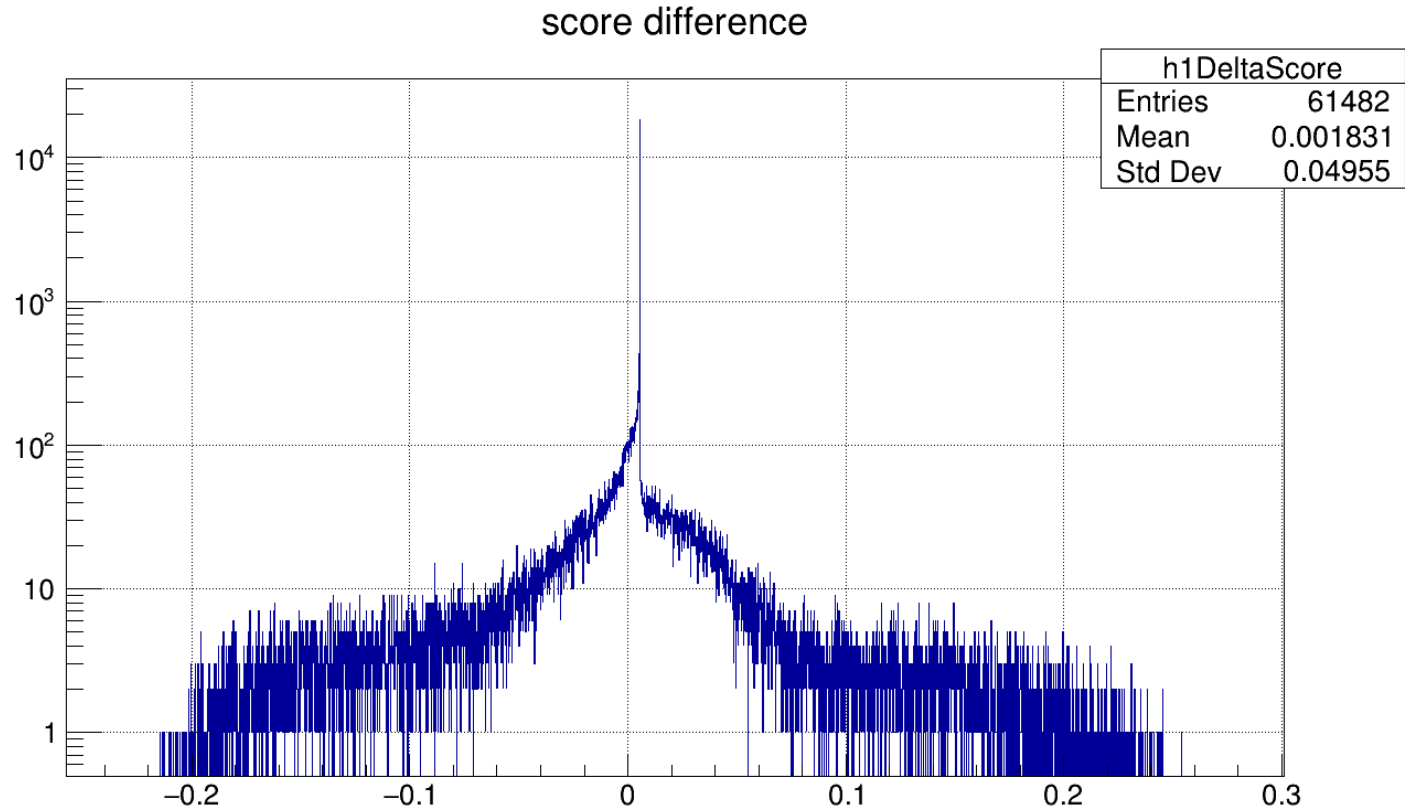
```
int compareScores(): checking 2 => 1: 0.000510988, 329.072, -0.112389, -2.99302, sig, 0.503355;
int compareScores(): comp. to 2 => 1: 0.000510988, 329.072, -0.112389, -2.99302, sig, 0.530041;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores():
int compareScores(): checking 3 => 1: 0.000510988, 173.786, 1.1569, 0.871613, sig, 0.396787;
int compareScores(): comp. to 3 => 1: 0.000510988, 173.786, 1.1569, 0.871613, sig, 0.386745;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores():
int compareScores(): checking 4 => 2: 0.000511107, 1195.96, 0.959032, 2.28129, sig, 0.551666;
int compareScores(): comp. to 4 => 2: 0.000510995, 184.375, 0.46453, -1.78823, sig, 0.418105;
int compareScores(): IS NOT SAME EVENT
int compareScores(): comp. to 4 => 2: 0.000511107, 1195.96, 0.959032, 2.28129, sig, 0.546264;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores():
int compareScores(): checking 4 => 2: 0.000510995, 184.375, 0.46453, -1.78823, sig, 0.409018;
int compareScores(): comp. to 4 => 2: 0.000510995, 184.375, 0.46453, -1.78823, sig, 0.418105;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores(): comp. to 4 => 2: 0.000511107, 1195.96, 0.959032, 2.28129, sig, 0.546264;
int compareScores(): IS NOT SAME EVENT
int compareScores():
int compareScores(): checking 5 => 1: 0.000511003, 143.726, -0.408077, 2.87403, sig, 0.402329;
int compareScores(): comp. to 5 => 1: 0.105658, 556.551, -0.30595, -1.90936, sig, 0.546237;
int compareScores(): IS NOT SAME EVENT
int compareScores():
int compareScores(): checking 7 => 1: 0.105658, 449.363, -2.25997, -0.0494479, sig, 0.514427;
int compareScores(): comp. to 7 => 1: 0.105658, 449.363, -2.25997, -0.0494479, sig, 0.543631;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores():
int compareScores(): checking 9 => 2: 0.000510988, 338.15, -0.826675, -1.26432, sig, 0.515486;
int compareScores(): comp. to 9 => 1: 0.000510988, 338.15, -0.826675, -1.26432, sig, 0.510905;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores():
int compareScores(): checking 9 => 2: 0.000511047, 156.798, -2.13055, 0.77728, sig, 0.363555;
int compareScores(): comp. to 9 => 1: 0.000510988, 338.15, -0.826675, -1.26432, sig, 0.510905;
int compareScores(): IS NOT SAME EVENT
int compareScores():
int compareScores(): checking 11 => 2: 0.000509672, 2418.06, 0.131391, -1.83342, sig, 0.551664;
int compareScores(): comp. to 11 => 2: 0.105658, 173.686, -1.44993, 3.04491, sig, 0.455407;
int compareScores(): IS NOT SAME EVENT
int compareScores(): comp. to 11 => 2: 0.000509672, 2418.06, 0.131391, -1.83342, sig, 0.546264;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores():
int compareScores(): checking 11 => 2: 0.105658, 173.686, -1.44993, 3.04491, sig, 0.421575;
int compareScores(): comp. to 11 => 2: 0.105658, 173.686, -1.44993, 3.04491, sig, 0.455407;
int compareScores(): IS SAME EVENT
int compareScores(): SCORES DO NOT MATCH
int compareScores():
int compareScores():
int compareScores():
```

Example of single event

Example of double event

Example of non-selected event (-99)

# Comparing scores



Here do the same  
plot for sig and  
bkg separately

**BONUS**

# An interesting observation

```
In [1]: import uproot3
import numpy as np
import pandas as pd

In [2]: DF_Signal = uproot3.open('../ntuples/original/Signal-0.root')['Nominal'].pandas.df()

In [3]: DF_Diboson = uproot3.open('../ntuples/original/Diboson-1.root')['Nominal'].pandas.df()
DF_Diboson.shape

Out[3]: (621719, 225)

In [4]: import sklearn.utils
def Shuffling(df):
    df = sklearn.utils.shuffle(df, random_state=123)
    df = df.reset_index(drop=True)
    return df

DF_Signal = Shuffling(DF_Signal)
DF_Diboson = Shuffling(DF_Diboson)

In [5]: DF_Signal.insert(len(DF_Signal.columns), 'isSignal', np.ones(DF_Signal.shape[0]), True)
DF_Diboson.insert(len(DF_Diboson.columns), 'isSignal', np.zeros(DF_Diboson.shape[0]), True)

signalList = [DF_Signal[:50000] ]
dibosonList = [DF_Diboson[:50000] ]

totalPD_sig = pd.concat(signalList, ignore_index=True)
totalPD_bkg = pd.concat(dibosonList, ignore_index=True)

print ('Signal events:', totalPD_sig.shape)
print ('Background events:', totalPD_bkg.shape)

df_total = pd.concat([totalPD_sig, totalPD_bkg], ignore_index=True)
df_total = Shuffling(df_total)
print ("Total events after shuffling: ", df_total.shape)

Signal events: (50000, 226)
Background events: (50000, 226)
Total events after shuffling: (100000, 226)

In [6]: InputFeatures = ['lep1_m', 'lep1_pt', 'lep1_eta', 'lep1_phi']#, 'lep2_m', 'lep2_pt', 'lep2_eta', 'lep2_phi', 'fatjet_
X=df_total[InputFeatures].values
y_tmp = df_total['isSignal']
```

If I don't do this, the DNN breaks down (!?)

