Lab of comp physics B

Usage of ICARUS data to identify eNeutrinos & Cosmic Ray

Proj supervisors: Filippo Varanini,
Christian Farnese & Alberto Zucchetta

Professor: Marco Zanetti

Ali Bavarchee
Vivek Janardhana

Exploring data

Step 1

Get the data & use uproot

The data was provided by the profs in form of .root files, we use ‘uproot=3.9’ and pandas to extract data to dataframes

Step 2

Clean data

Removal of NA values, excluding .root files not in format of other files

Done for both eNeutrino data and Cosmic rays

Put into uniform format data frame

Step 3

Plot the data

For each .root file

1 subentry = 1 signal

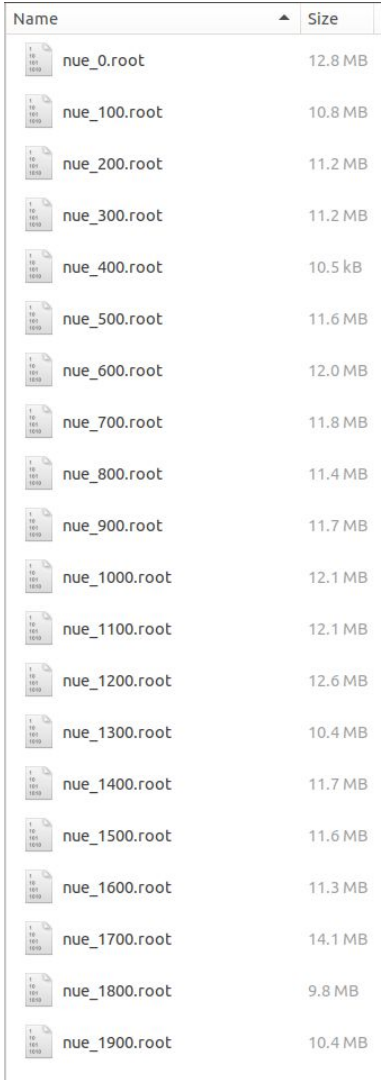
1 entry = 1 event

Plot 2D plot with hit_peak=X hit_wire=Y

Use the plot as input to ML

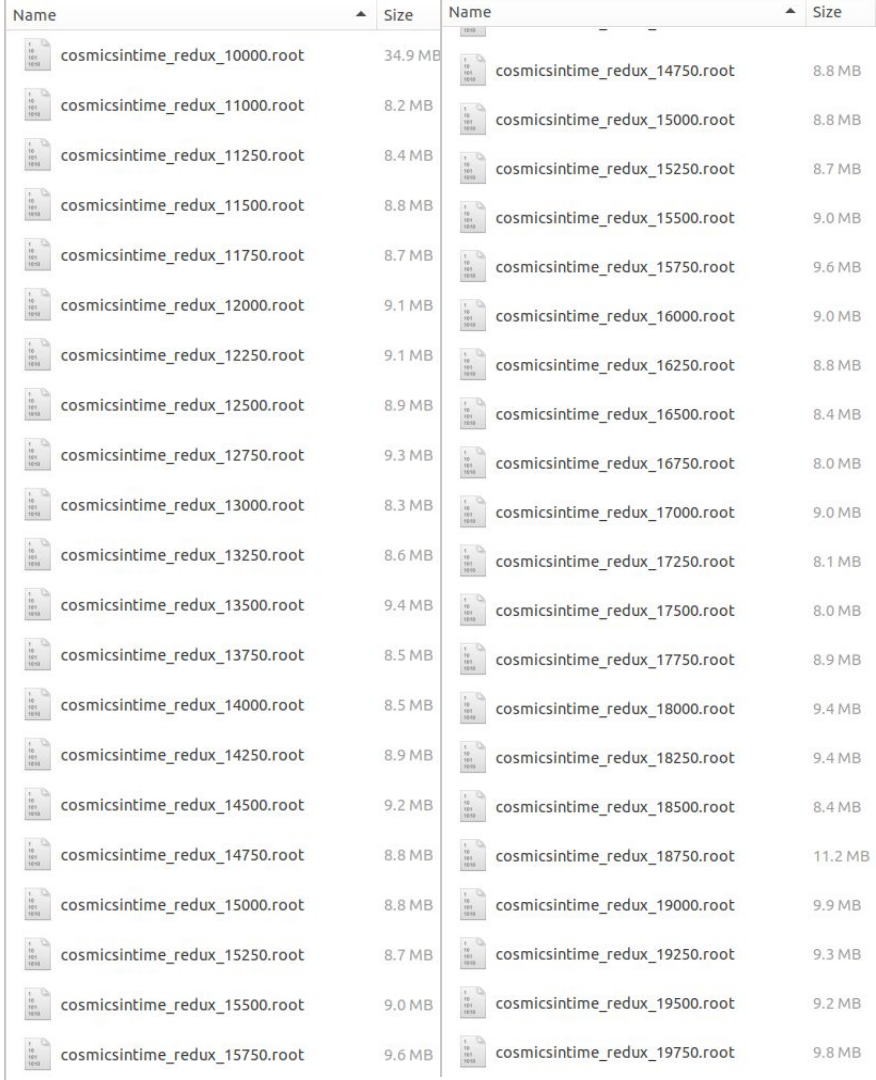
Step 1

eNu .root Files



Name	Size
nue_0.root	12.8 MB
nue_100.root	10.8 MB
nue_200.root	11.2 MB
nue_300.root	11.2 MB
nue_400.root	10.5 kB
nue_500.root	11.6 MB
nue_600.root	12.0 MB
nue_700.root	11.8 MB
nue_800.root	11.4 MB
nue_900.root	11.7 MB
nue_1000.root	12.1 MB
nue_1100.root	12.1 MB
nue_1200.root	12.6 MB
nue_1300.root	10.4 MB
nue_1400.root	11.7 MB
nue_1500.root	11.6 MB
nue_1600.root	11.3 MB
nue_1700.root	14.1 MB
nue_1800.root	9.8 MB
nue_1900.root	10.4 MB

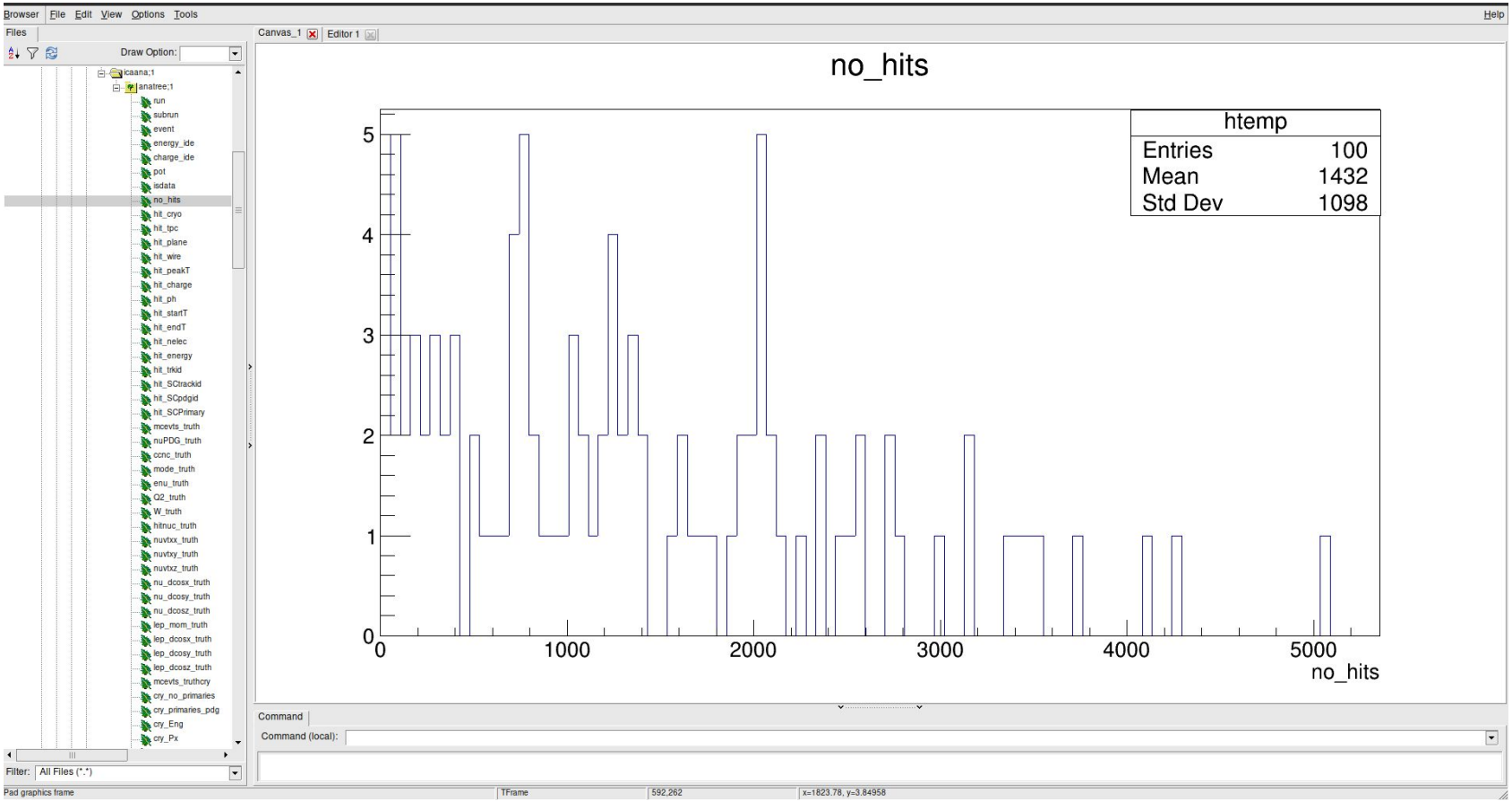
cosmic Nu .root Files

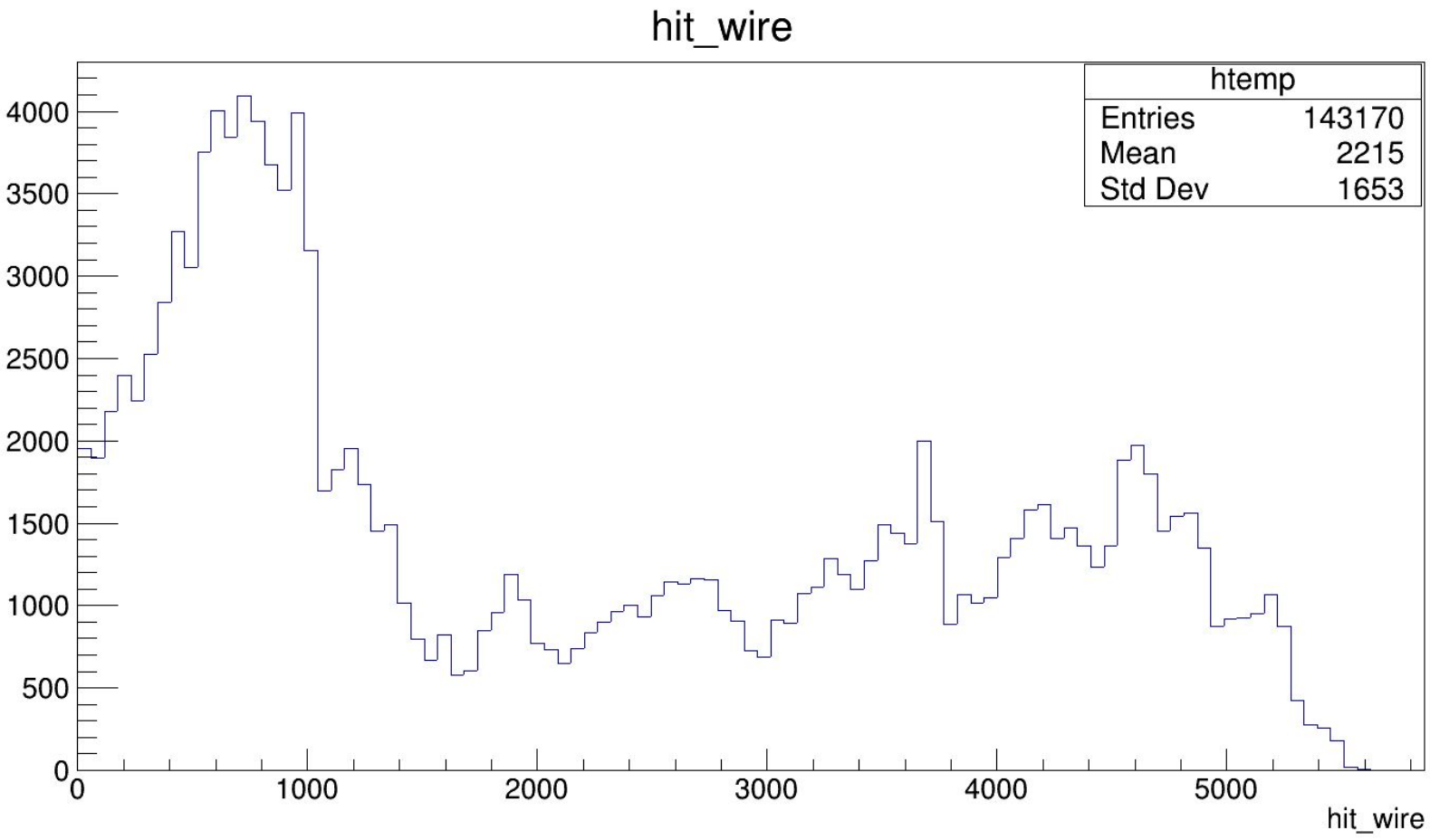


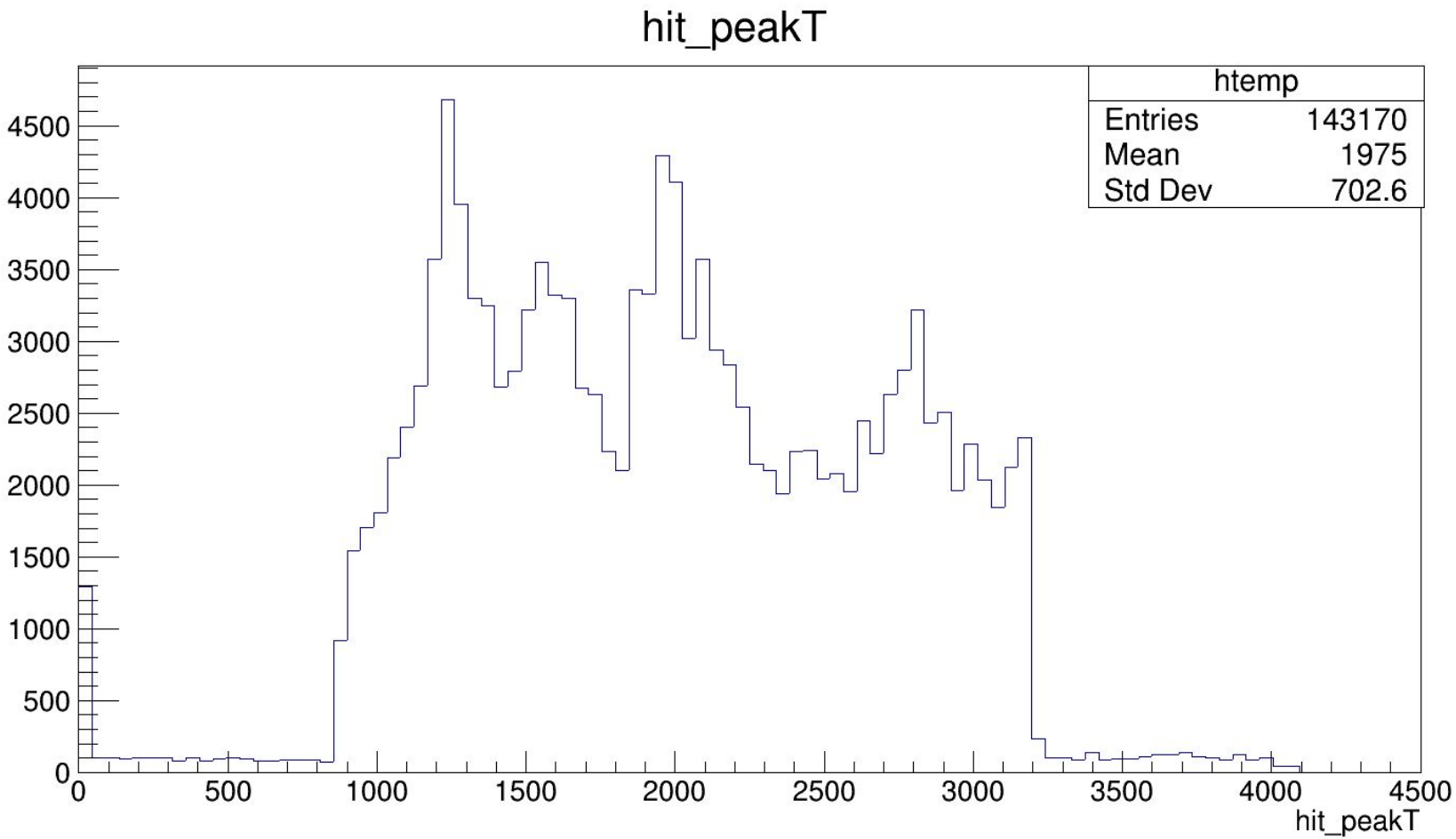
Name	Size
cosmicsintime_redux_10000.root	34.9 MB
cosmicsintime_redux_11000.root	8.2 MB
cosmicsintime_redux_11250.root	8.4 MB
cosmicsintime_redux_11500.root	8.8 MB
cosmicsintime_redux_11750.root	8.7 MB
cosmicsintime_redux_12000.root	9.1 MB
cosmicsintime_redux_12250.root	9.1 MB
cosmicsintime_redux_12500.root	8.9 MB
cosmicsintime_redux_12750.root	9.3 MB
cosmicsintime_redux_13000.root	8.3 MB
cosmicsintime_redux_13250.root	8.6 MB
cosmicsintime_redux_13500.root	9.4 MB
cosmicsintime_redux_13750.root	8.5 MB
cosmicsintime_redux_14000.root	8.5 MB
cosmicsintime_redux_14250.root	8.9 MB
cosmicsintime_redux_14500.root	9.2 MB
cosmicsintime_redux_14750.root	8.8 MB
cosmicsintime_redux_15000.root	8.8 MB
cosmicsintime_redux_15250.root	8.7 MB
cosmicsintime_redux_15500.root	9.0 MB
cosmicsintime_redux_15750.root	9.6 MB
cosmicsintime_redux_16000.root	9.0 MB
cosmicsintime_redux_16250.root	8.8 MB
cosmicsintime_redux_16500.root	8.4 MB
cosmicsintime_redux_16750.root	8.0 MB
cosmicsintime_redux_17000.root	9.0 MB
cosmicsintime_redux_17250.root	8.1 MB
cosmicsintime_redux_17500.root	8.0 MB
cosmicsintime_redux_17750.root	8.9 MB
cosmicsintime_redux_18000.root	9.4 MB
cosmicsintime_redux_18250.root	9.4 MB
cosmicsintime_redux_18500.root	8.4 MB
cosmicsintime_redux_18750.root	11.2 MB
cosmicsintime_redux_19000.root	9.9 MB
cosmicsintime_redux_19250.root	9.3 MB
cosmicsintime_redux_19500.root	9.2 MB
cosmicsintime_redux_19750.root	9.8 MB

Root hists:

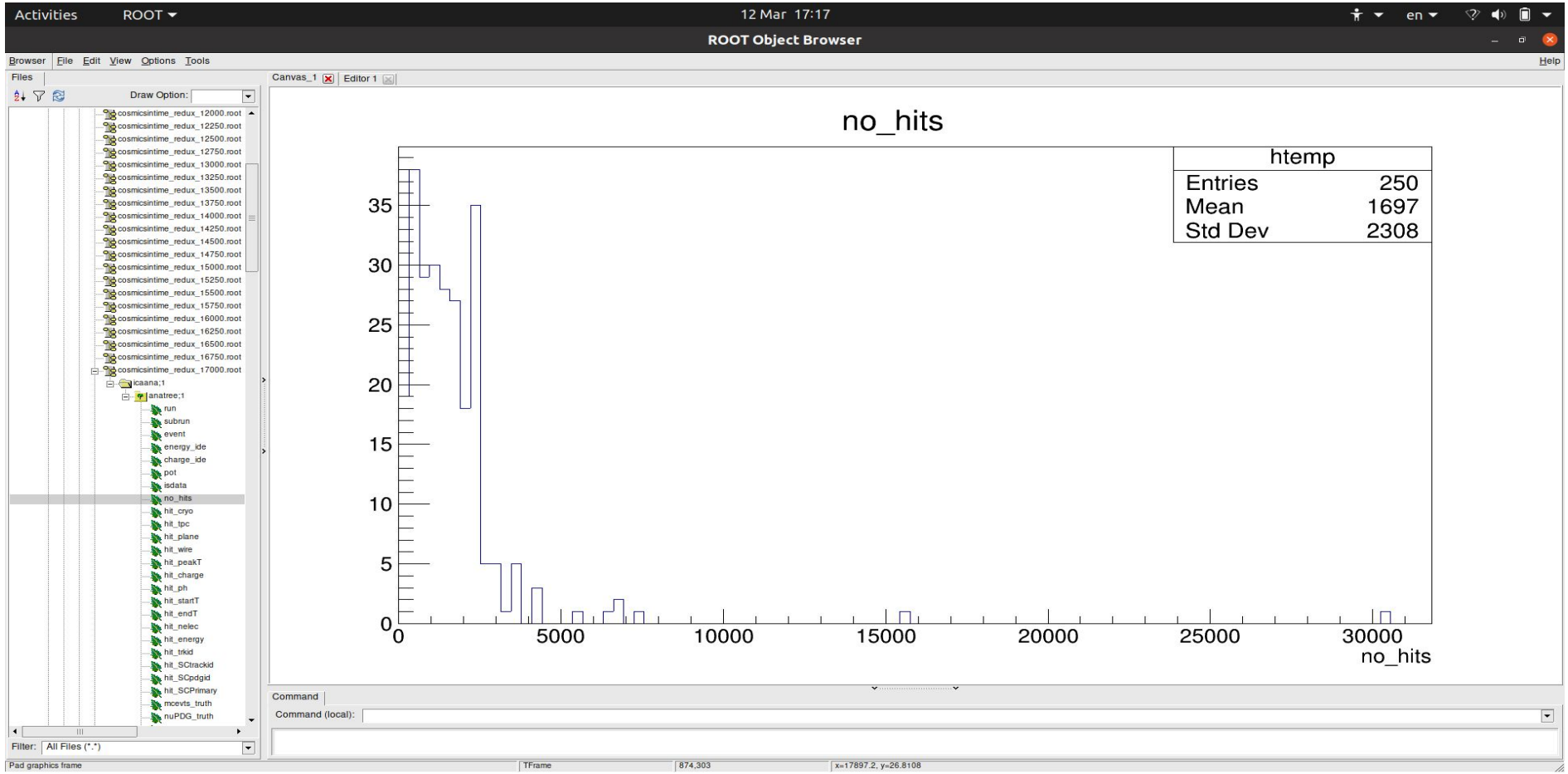
nue_1700.root:

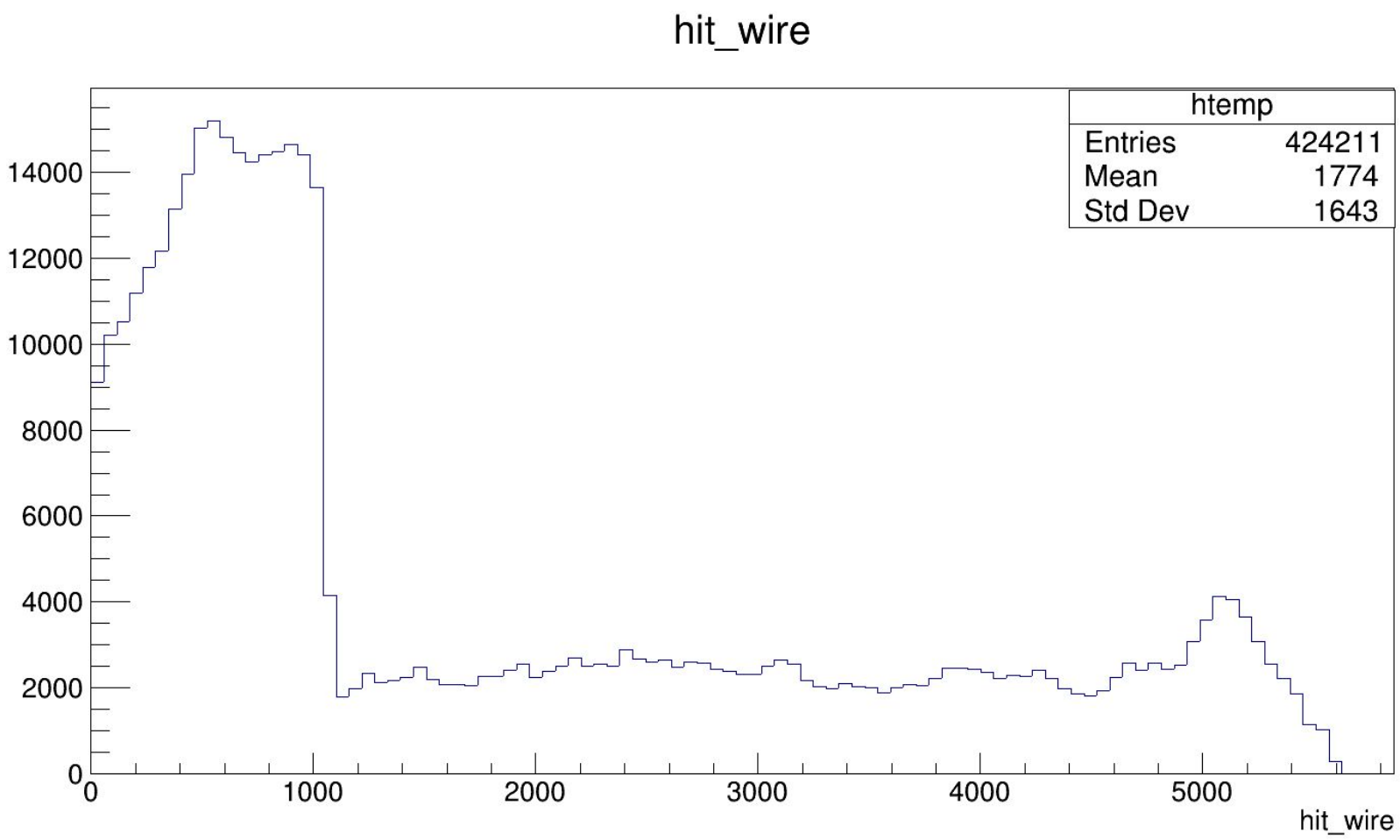


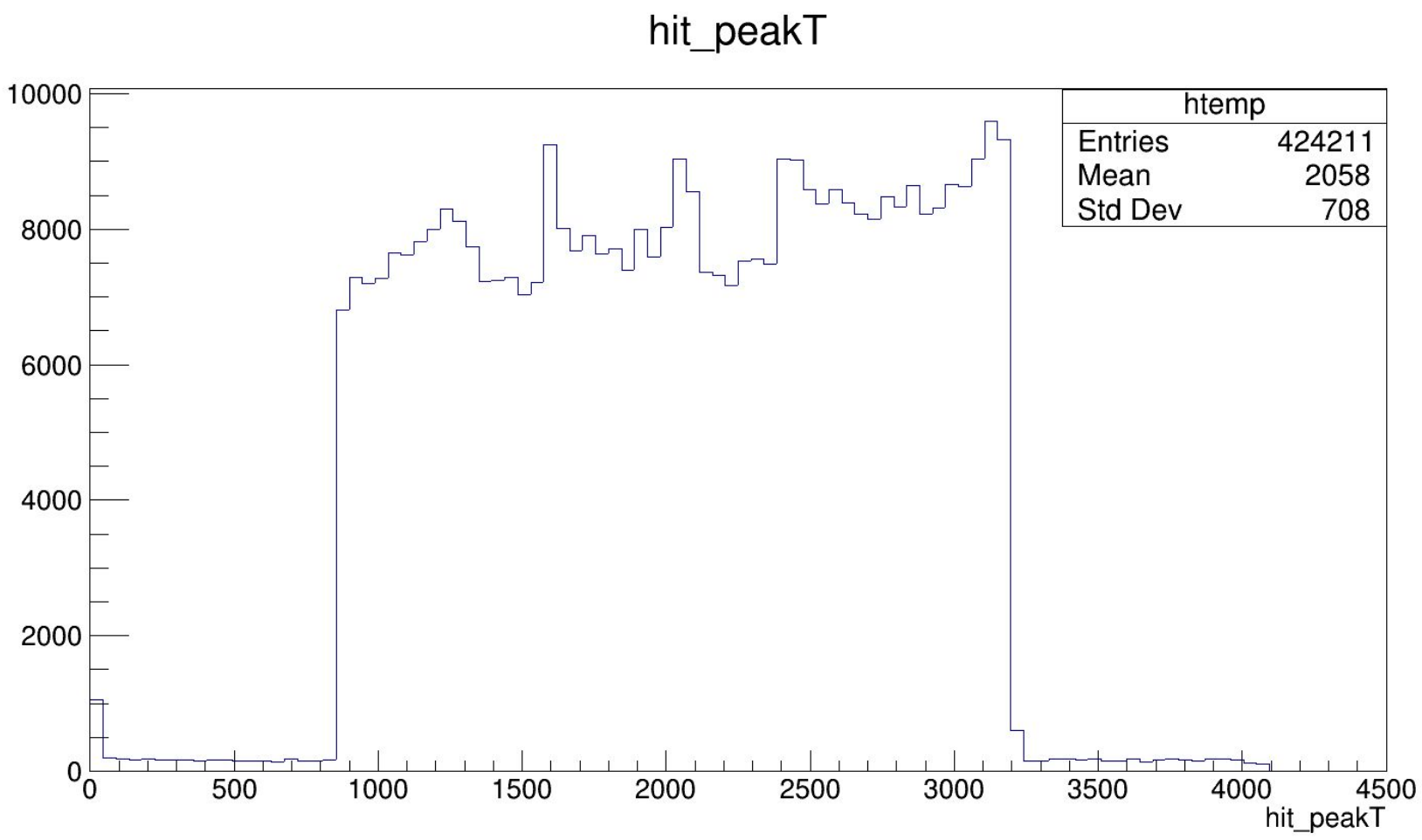

hit_wire
htemp
Entries 143170
Mean 2215
Std Dev 1653
4000
3500
3000
2500
2000
1500
1000
500
0
0
1000
2000
3000
4000
5000
hit_wire


hit_peakT
htemp
Entries 143170
Mean 1975
Std Dev 702.6
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
hit_peakT

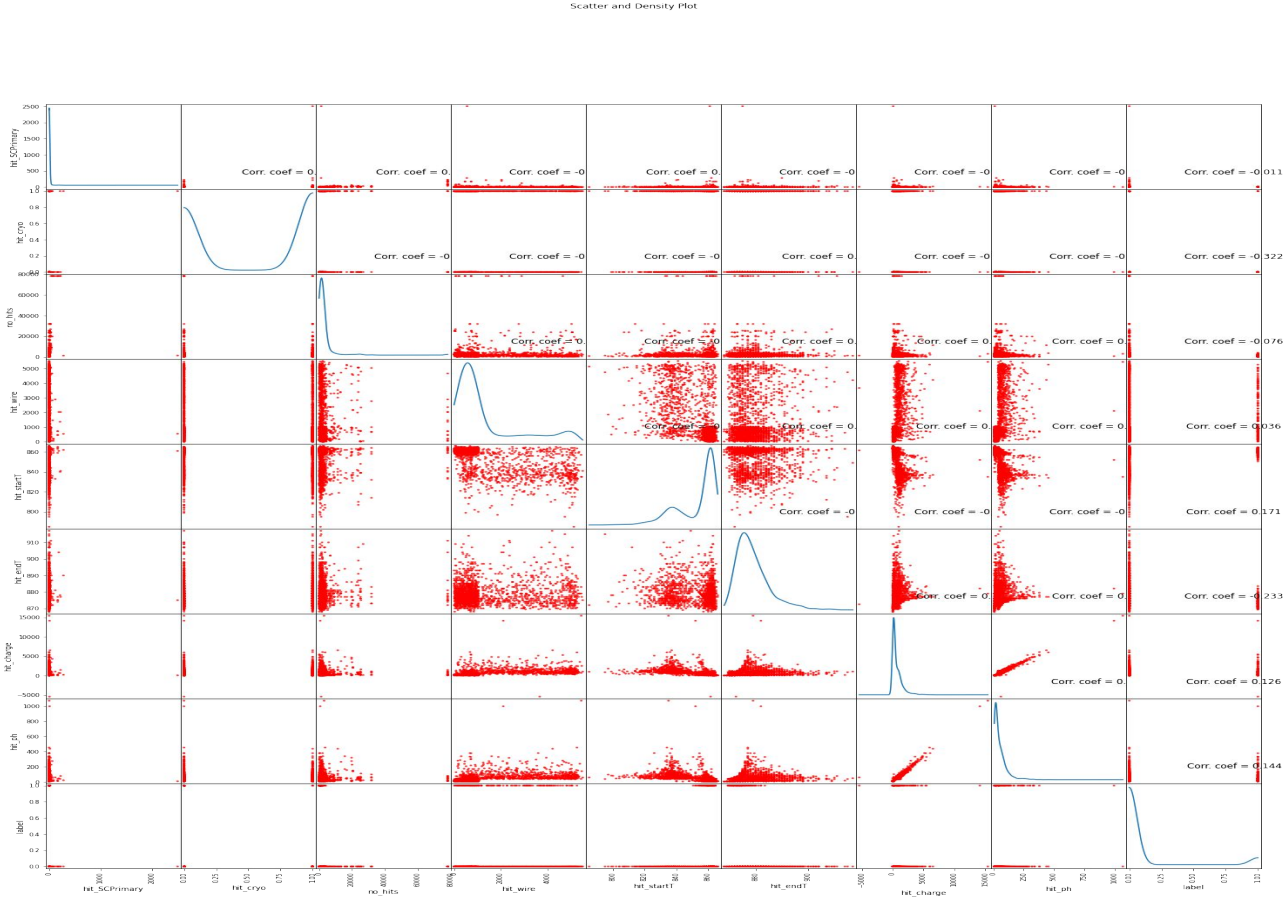
cosmicsintime_redux_17000.root:



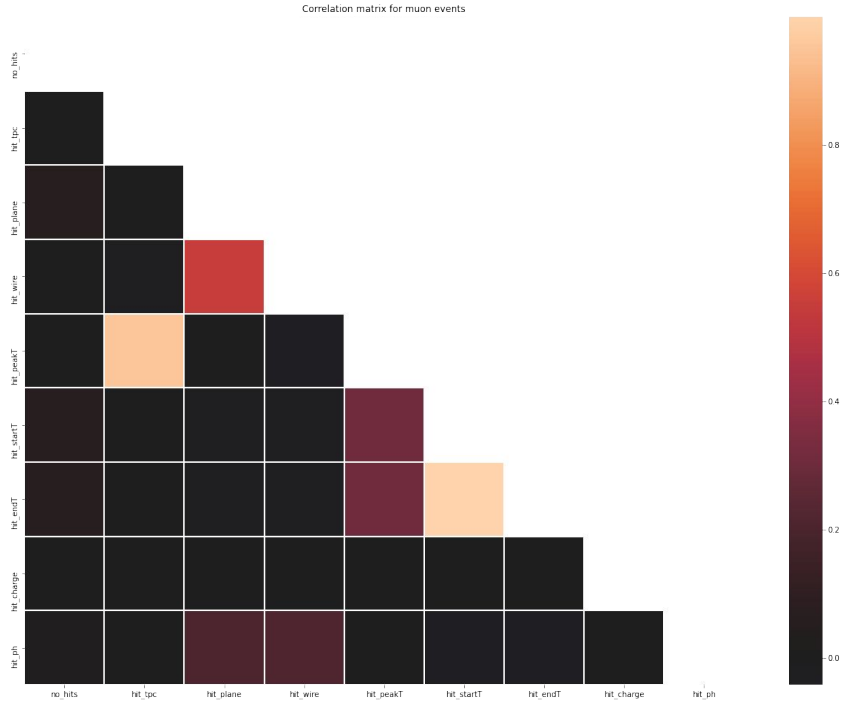

hit_wire
htemp
Entries 424211
Mean 1774
Std Dev 1643
14000
12000
10000
8000
6000
4000
2000
0
0
1000
2000
3000
4000
5000
hit_wire

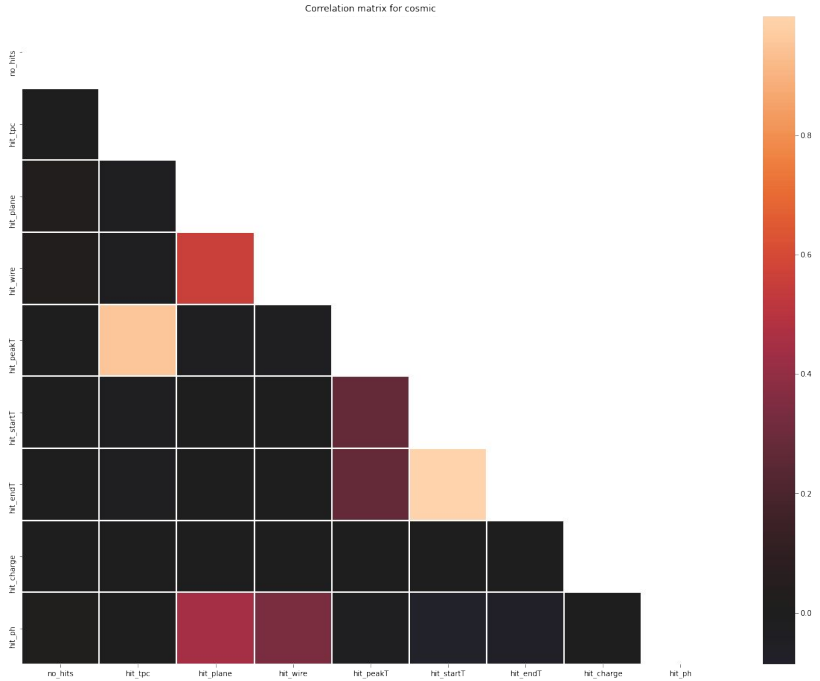

hit_peakT
htemp
Entries 424211
Mean 2058
Std Dev 708
10000
8000
6000
4000
2000
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
hit_peakT

Correlation Matrix



Import the data and select the features:





Data Mining

The two set of data were collected as "nue" and "cosmicsintime_redux" represent SBN and cosmic ray. The files are simulation/ true data in .root type. The main goal is training an algorithm to discriminate these two sets of data. Data has been imported to python by Uproot and formed a data frame by pandas library.

The next job is the selection of the most relevant features.

```
1 colmns = ['no_hits','hit_tpc', 'hit_plane','hit_wire', 'hit_peakT']
2 print('features: ',colmns)
```

```
features:  ['no_hits', 'hit_tpc', 'hit_plane', 'hit_wire', 'hit_peakT']
```

Step 2

Data Cleaning and Preparation

```
1  df_con = df_con.dropna()
2  print(df_con.info())
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 2276478 entries, ('./nue_0.root', 0, 0) to ('./nue_900.root', 99, 332)
Data columns (total 7 columns):
 #   Column     Dtype
---  ------     -----
 0   hit_tpc    int16
 1   hit_plane  int16
 2   no_hits    int32
 3   hit_wire   int16
 4   hit_peakT  float32
 5   label      int64
 6   tagg       object
dtypes: float32(1), int16(3), int32(1), int64(1), object(1)
memory usage: 74.0+ MB
None
```

Merging cosmic df and nue df

```
data = pd.concat([df_in0_2, dfc_in0_2])
data
```

			no_hits	hit_wire	hit_peakT	label
	entry	subentry				
./nue_0.root	0	0	29	20	2147.000000	1
		1	29	34	3996.000000	1
		2	29	43	551.000000	1
		3	29	64	2683.000000	1
		4	29	65	2683.000000	1
...	...	...	...	...	...	...
./cosmicsintime_redux_19750.root	249	147	1424	5082	3178.193848	0
		148	1424	5083	3184.908447	0
		149	1424	5084	3192.386230	0
		150	1424	5085	3192.533691	0
		151	1424	5099	3506.603760	0

2588127 rows × 4 columns

Step 3

The hits of Collection from TPC 0

nues:

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
./nue_0.root	0	25	0	2	29	975	2093.201660	1	ev
		26	0	2	29	976	2083.646240	1	ev
		27	0	2	29	983	2683.135254	1	ev
		28	0	2	29	984	2683.608398	1	ev
	1	420	0	2	29	337	3101.061523	1	ev
...	...	...	...	...	...	...	...	...	...
./nue_900.root	99	328	0	2	333	378	559.562134	1	ev
		329	0	2	333	398	869.550354	1	ev
		330	0	2	333	399	873.309265	1	ev
		331	0	2	333	478	2135.397461	1	ev
		332	0	2	333	479	2137.460693	1	ev

392974 rows × 7 columns

cosmic df:

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
./cosmicsintime_redux_10000.root	**0**	**6**	0	2	176	4920	2576.753662	0	cos
		7	0	2	176	4921	2577.204102	0	cos
		8	0	2	176	4972	794.500000	0	cos
	1	**749**	0	2	2343	1059	3193.472412	0	cos
		750	0	2	2343	1060	3193.355713	0	cos
...	**...**	**...**	...	...	...	...	...	...	...
./cosmicsintime_redux_19750.root	**249**	**147**	0	2	1424	5082	3178.193848	0	cos
		148	0	2	1424	5083	3184.908447	0	cos
		149	0	2	1424	5084	3192.386230	0	cos
		150	0	2	1424	5085	3192.533691	0	cos
		151	0	2	1424	5099	3506.603760	0	cos

2285561 rows × 7 columns

Concatenate nue and cosmic data frame:

	entry	subentry	no_hits	hit_wire	hit_peakT	label
./nue_0.root	**1**	**812**	1369	346	2944.0	1
		813	1369	357	2986.0	1
		814	1369	358	2992.0	1
		815	1369	361	1436.0	1
		816	1369	784	1946.0	1
...	**...**	**...**	...	...	...	...
./cosmicsintime_redux_19750.root	**249**	**807**	1424	5427	889.0	0
		808	1424	5428	883.0	0
		809	1424	5429	876.0	0
		810	1424	5430	869.0	0
		811	1424	5438	2231.0	0

2791403 rows × 4 columns

The hits of Induction 2 from TPC 1:

nues df:

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
./nue_0.root	1	812	1	1	1369	346	2944.0	1	ev
		813	1	1	1369	357	2986.0	1	ev
		814	1	1	1369	358	2992.0	1	ev
		815	1	1	1369	361	1436.0	1	ev
		816	1	1	1369	784	1946.0	1	ev
...	...	...	...	...	...	...	...	...	...
./nue_900.root	98	1335	1	1	2229	5467	2971.0	1	ev
		1336	1	1	2229	5468	2981.0	1	ev
		1337	1	1	2229	5469	2986.0	1	ev
		1338	1	1	2229	5483	2677.0	1	ev
		1339	1	1	2229	5486	2252.0	1	ev

406378 rows × 7 columns

cosmic df:

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
./cosmicsintime_redux_10000.root	**0**	**76**	1	1	176	5095	1925.0	0	cos
		77	1	1	176	5101	2717.0	0	cos
		78	1	1	176	5116	1054.0	0	cos
		79	1	1	176	5122	1076.0	0	cos
		80	1	1	176	5128	2391.0	0	cos
...	**...**	**...**	...	...	...	...	...	...	...
./cosmicsintime_redux_19750.root	**249**	**807**	1	1	1424	5427	889.0	0	cos
		808	1	1	1424	5428	883.0	0	cos
		809	1	1	1424	5429	876.0	0	cos
		810	1	1	1424	5430	869.0	0	cos
		811	1	1	1424	5438	2231.0	0	cos

2385025 rows × 7 columns

Concatenate nue and cosmic data frame:

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
./nue_0.root	**1**	**812**	1	1	1369	346	2944.0	1	ev
		813	1	1	1369	357	2986.0	1	ev
		814	1	1	1369	358	2992.0	1	ev
		815	1	1	1369	361	1436.0	1	ev
		816	1	1	1369	784	1946.0	1	ev
...	**...**	**...**	...	...	...	...	...	...	...
./cosmicsintime_redux_19750.root	**249**	**807**	1	1	1424	5427	889.0	0	cos
		808	1	1	1424	5428	883.0	0	cos
		809	1	1	1424	5429	876.0	0	cos
		810	1	1	1424	5430	869.0	0	cos
		811	1	1	1424	5438	2231.0	0	cos

2791403 rows × 7 columns

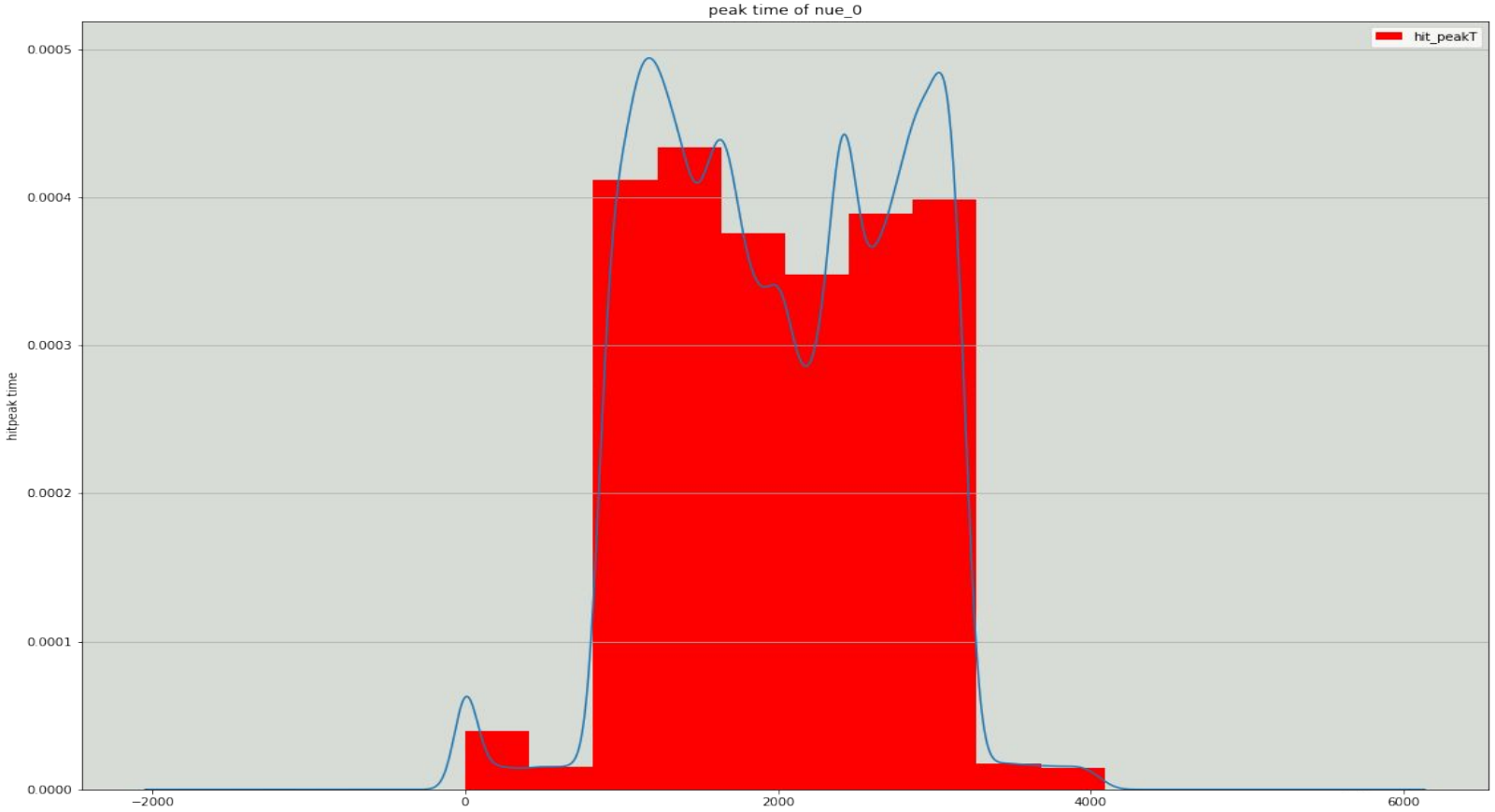
Hits of merging the TPCs

Input df:

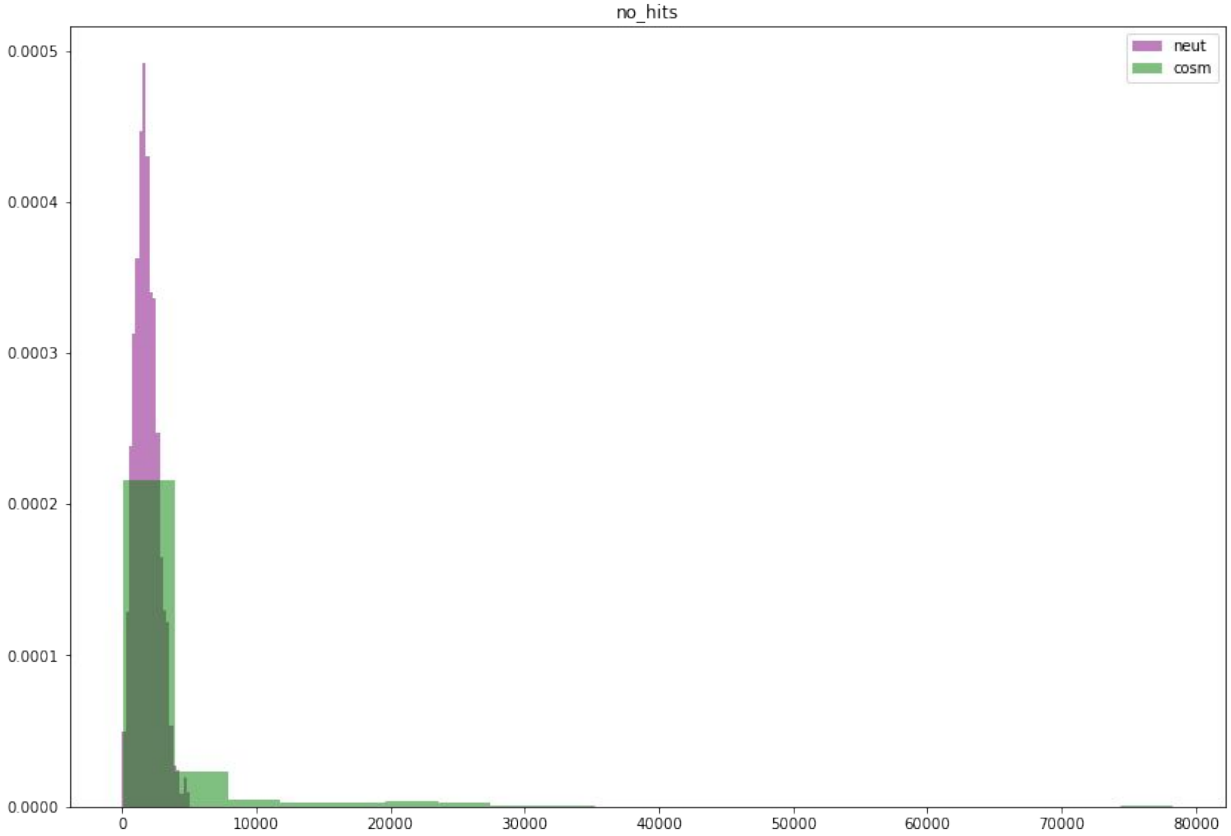
			hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
	entry	subentry							
./nue_0.root	0	25	0	2	29	975	1228.201660	1	ev
		26	0	2	29	976	1218.646240	1	ev
		27	0	2	29	983	1818.135254	1	ev
		28	0	2	29	984	1818.608398	1	ev
	1	420	0	2	1369	337	2236.061523	1	ev
...	...	...	...	...	...	...	...	...	...
./cosmicsintime_redux_19750.root	249	807	1	1	1424	5427	4616.000000	0	cos
		808	1	1	1424	5428	4622.000000	0	cos
		809	1	1	1424	5429	4629.000000	0	cos
		810	1	1	1424	5430	4636.000000	0	cos
		811	1	1	1424	5438	3274.000000	0	cos

5469938 rows × 7 columns

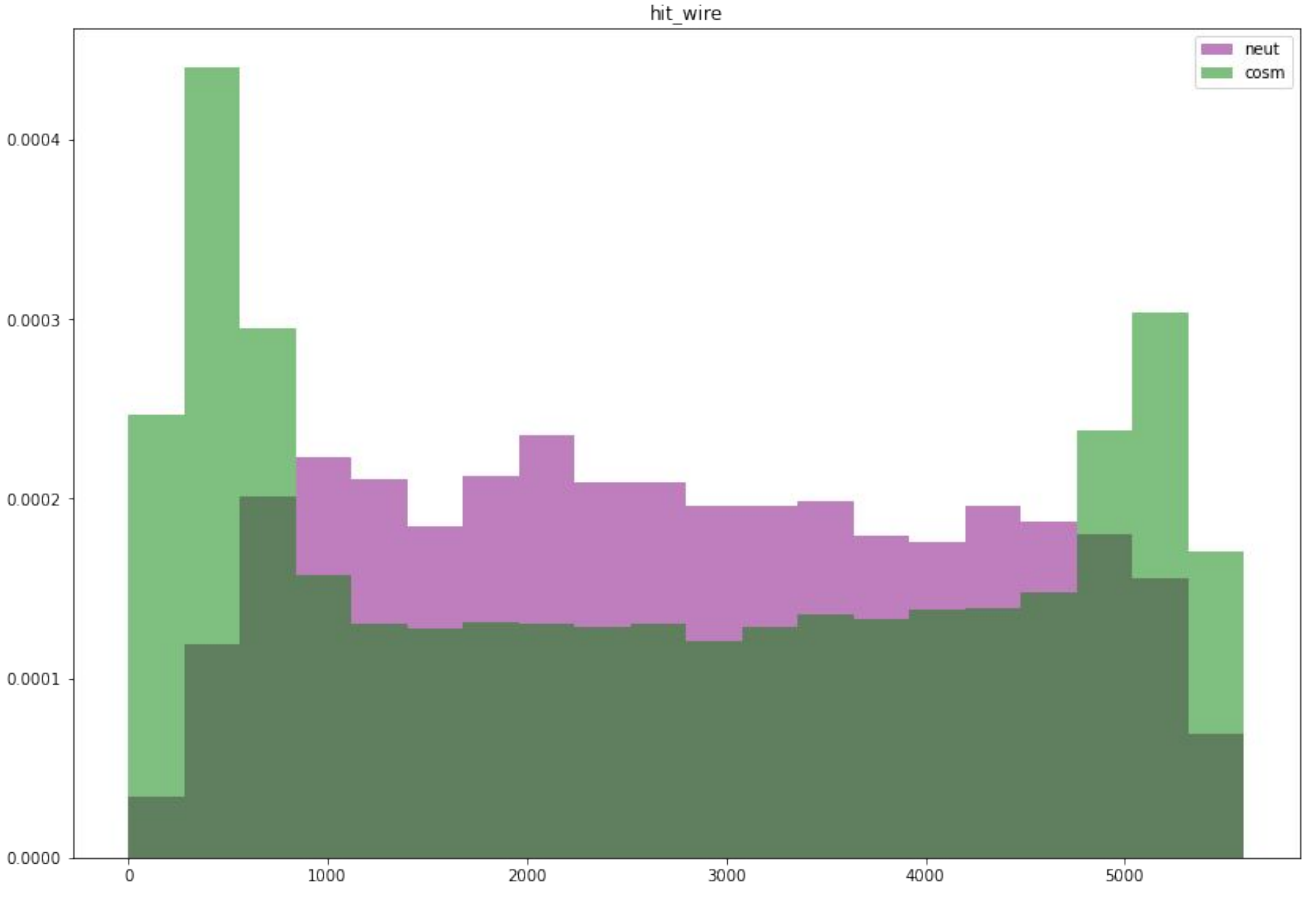
Examples of pyhist:



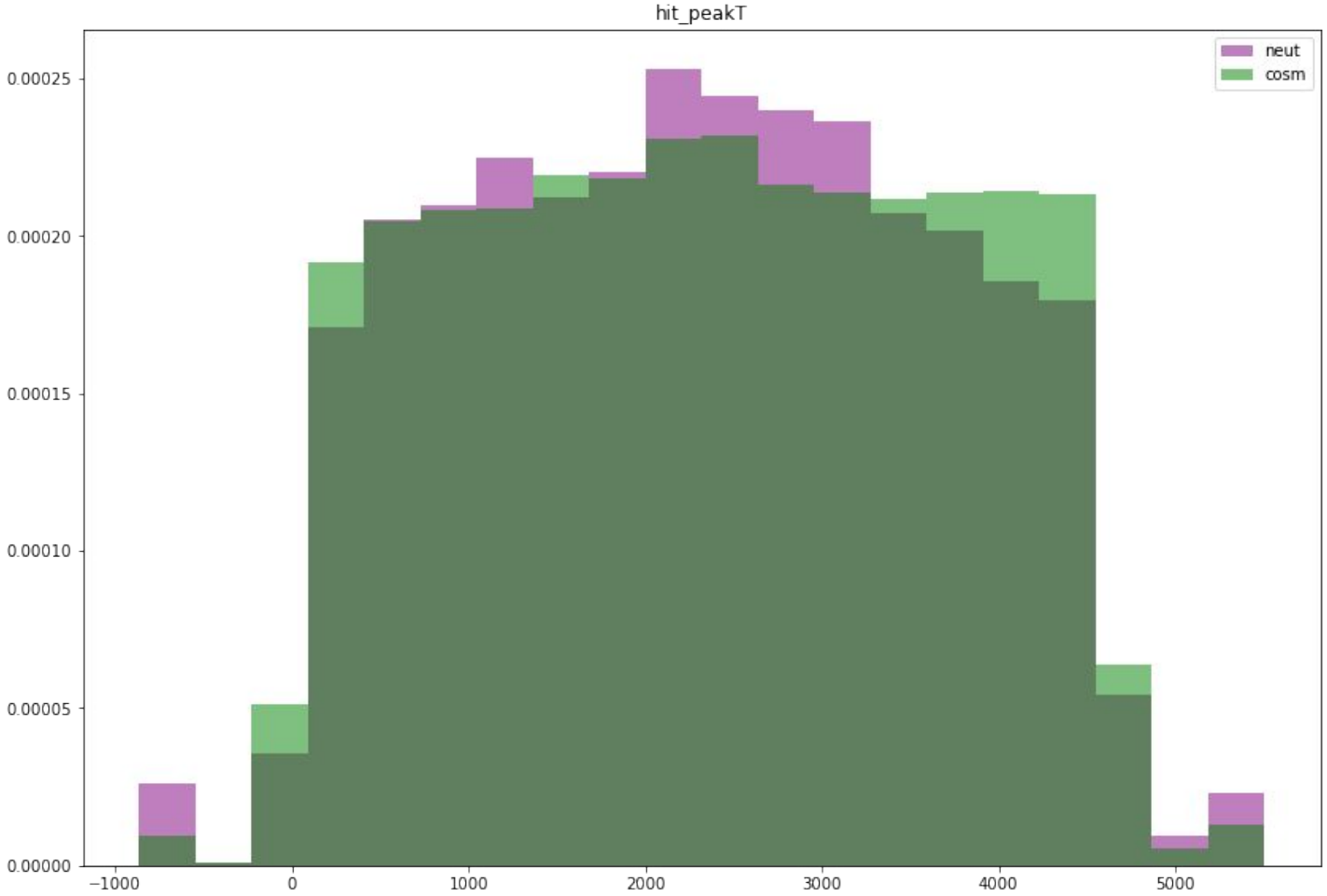
Python Hists: Combined area



Combined area

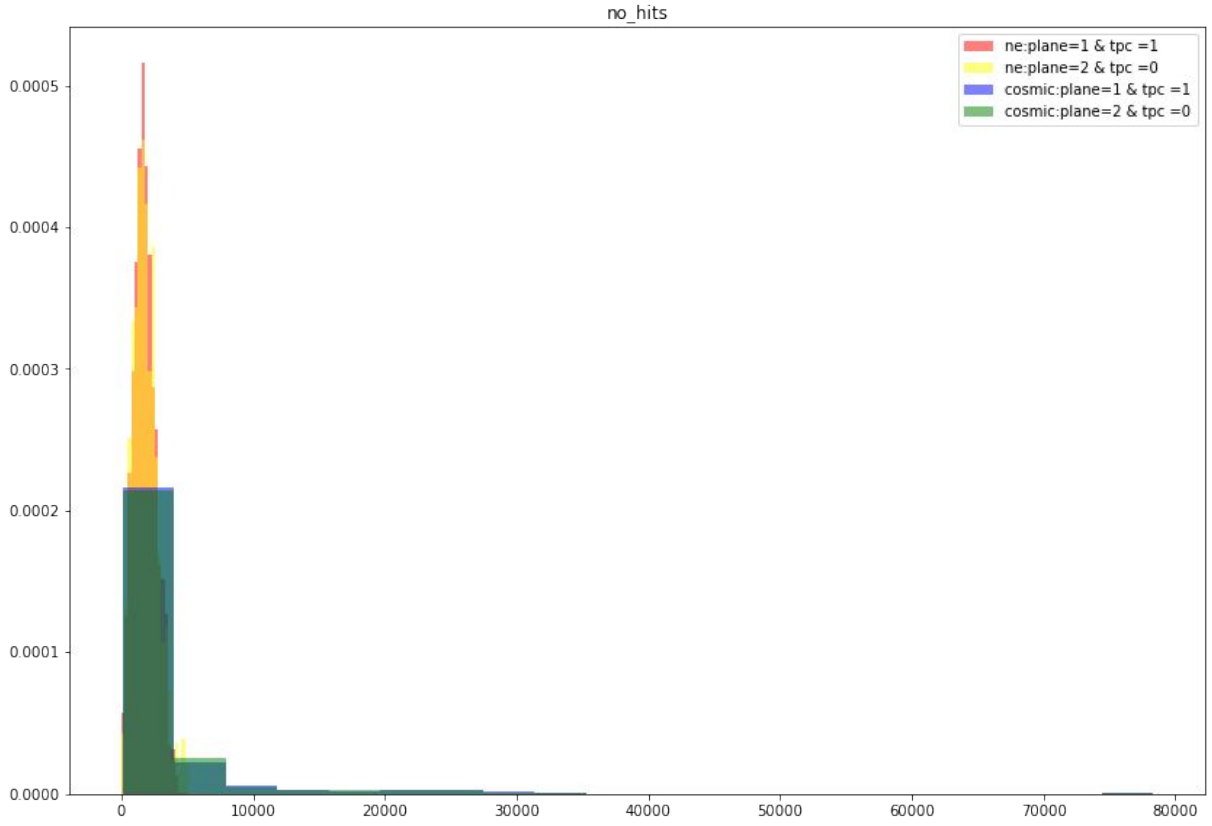


Combined area

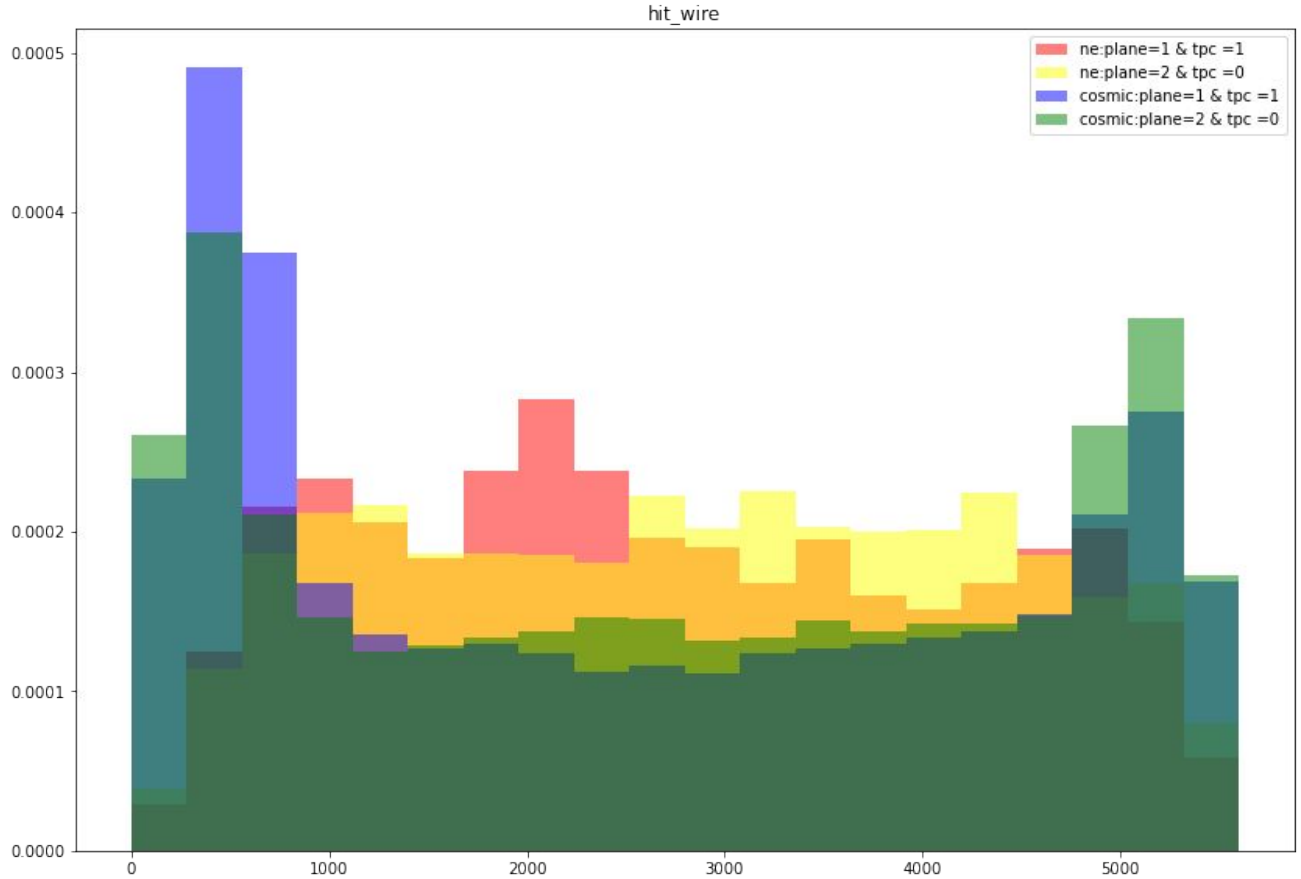


Combined area

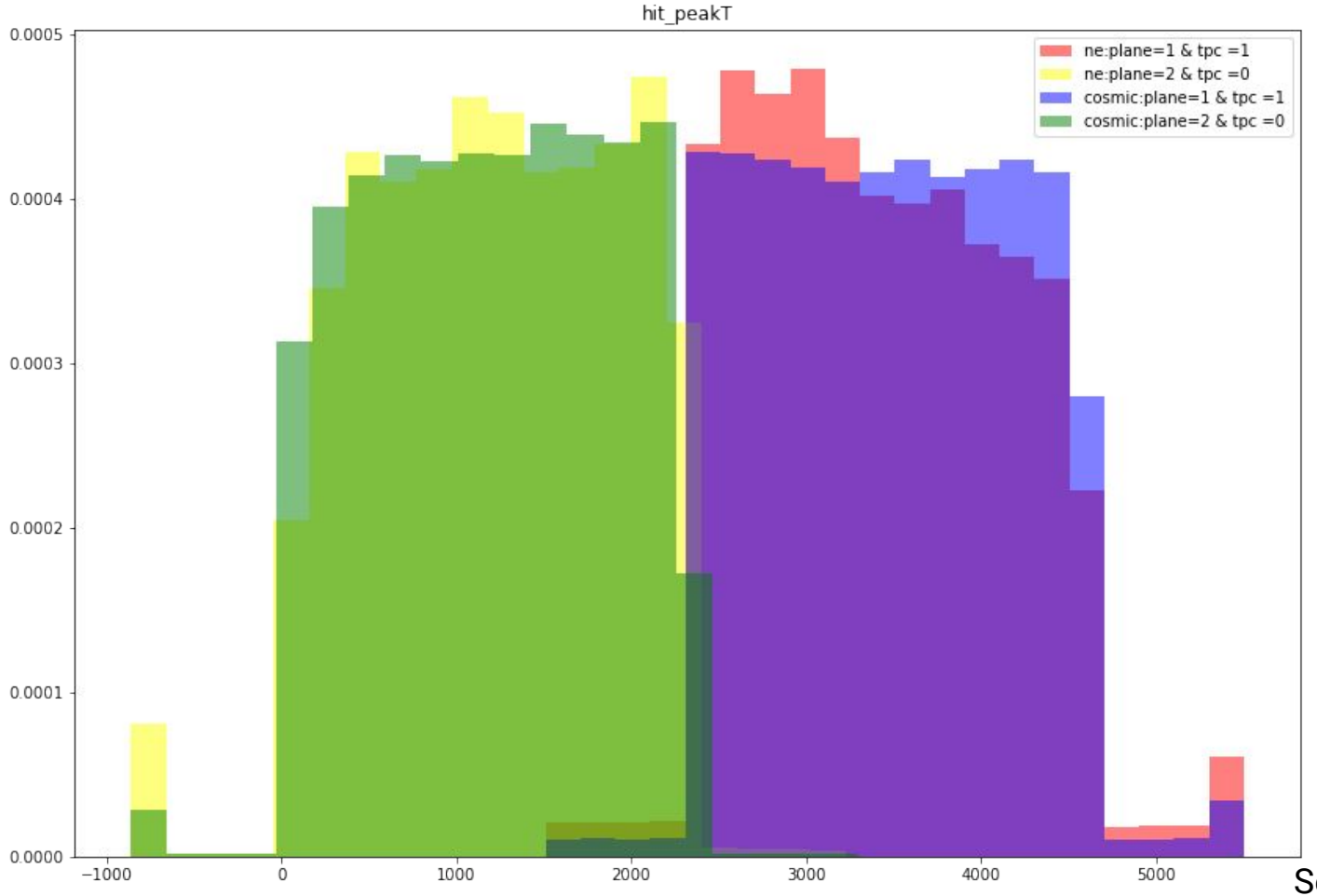
Py hists: Segregated areas



Segregated areas

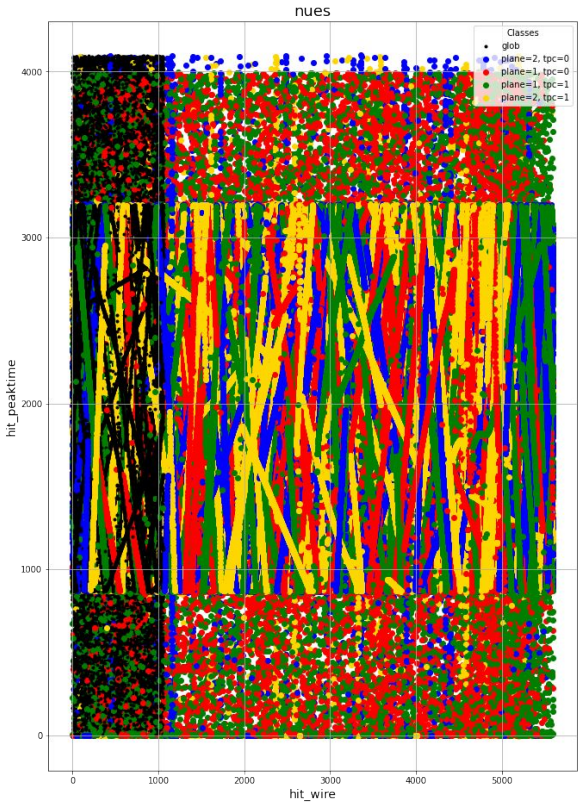


Segregated areas

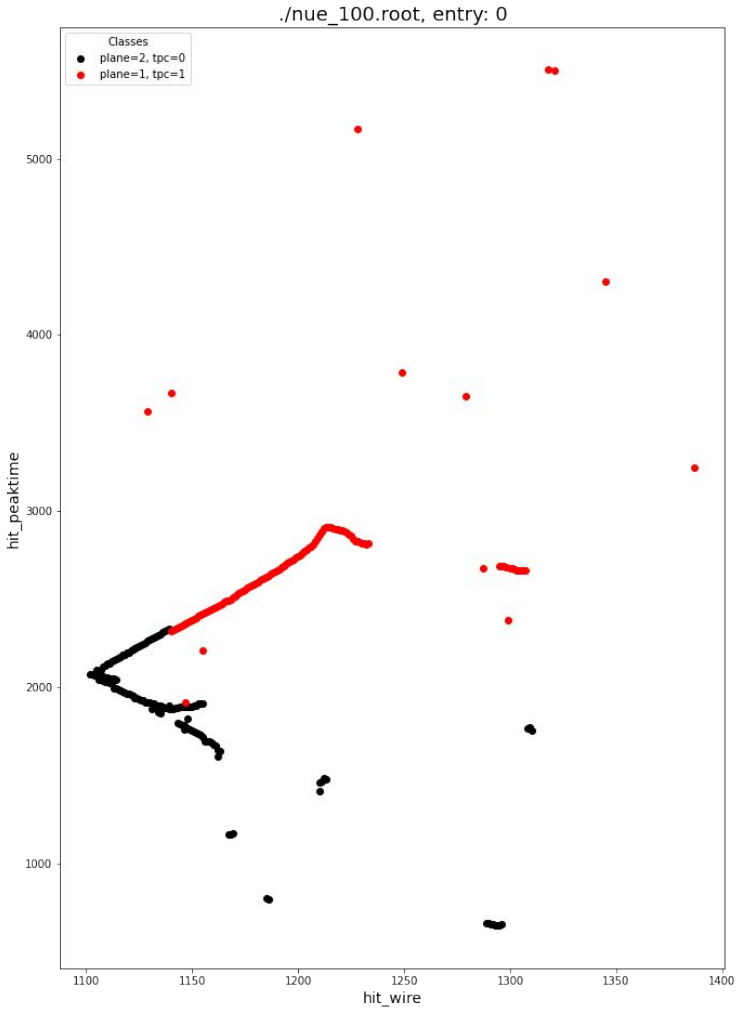


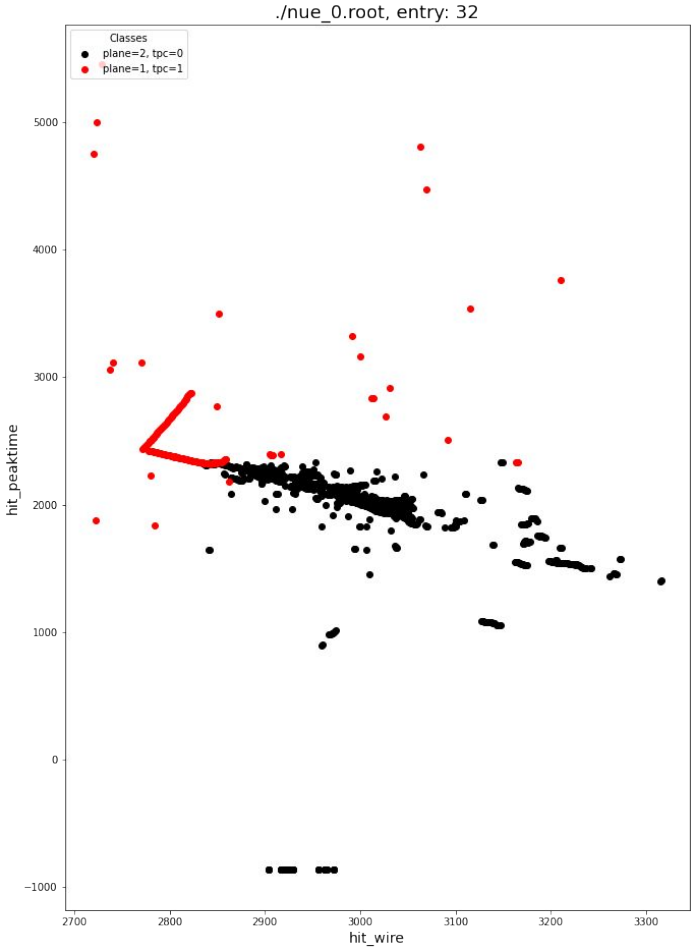
Segregated areas

Plots of nue.root files



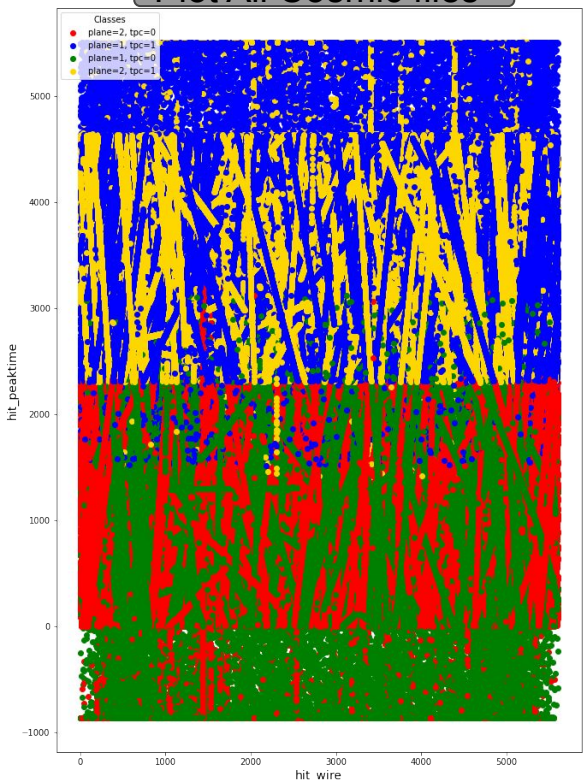
https://colab.research.google.com/drive/10W6nnlcXSQu3JUweu96fpHeoHKae5yiT#scrollTo=KeIBKnM5zIqy&line=2&uniqifier=1


./nue_100.root, entry: 0
Classes
plane=2, tpc=0
plane=1, tpc=1
hit_peaktime
hit_wire
5000
4000
3000
2000
1000
1100
1150
1200
1250
1300
1350
1400


./nue_0.root, entry: 32
Classes
plane=2, tpc=0
plane=1, tpc=1
hit_peaktime
hit_wire
5000
4000
3000
2000
1000
0
−1000
2700
2800
2900
3000
3100
3200
3300

Plot of cosmicsintime.root files

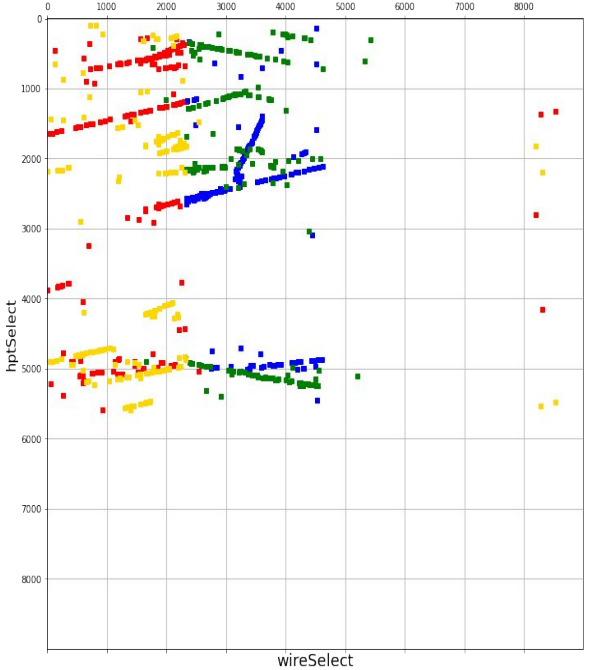
Plot All Cosmic files



https://colab.research.google.com/drive/1WxA_HGpU-ZpIASoOyU4IMTqA50K83uOJ#scrollTo=LrvO0bZF3NXh&line=3&uniqifier=1

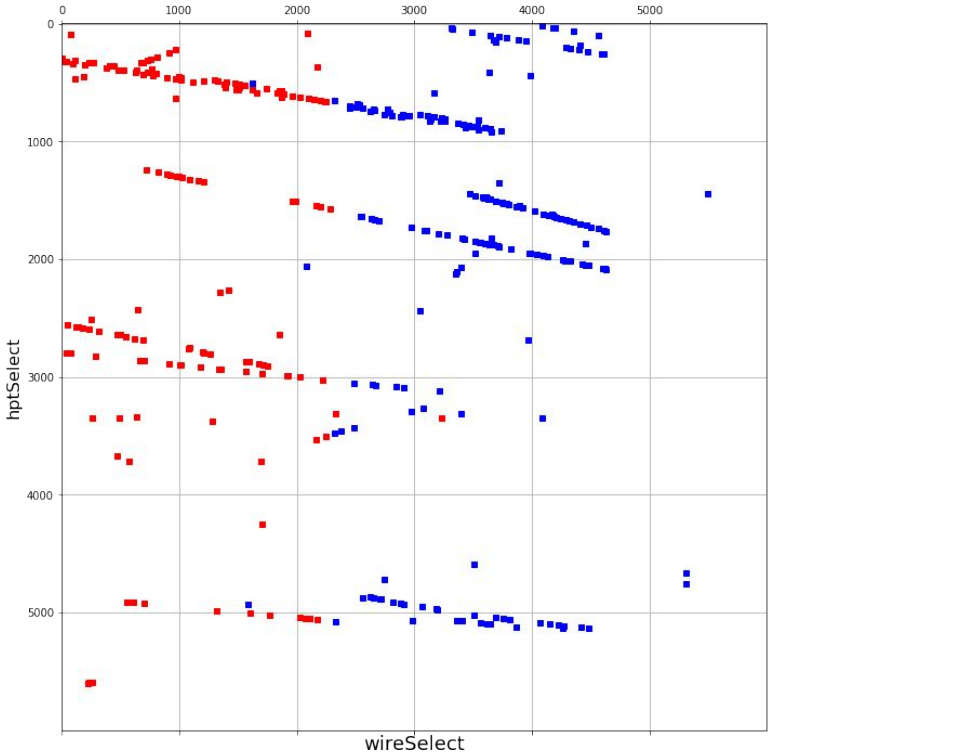
Plot of cosmicsintime.root files

./cosmicsintime_redux_13000.root plane = 1 , tpc = 0 (red) ||| plane = 2, tpc = 1 (blue) ||| plane = 1 , tpc = 1 (geen) ||| plane = 2, tpc = 0 (yellow)



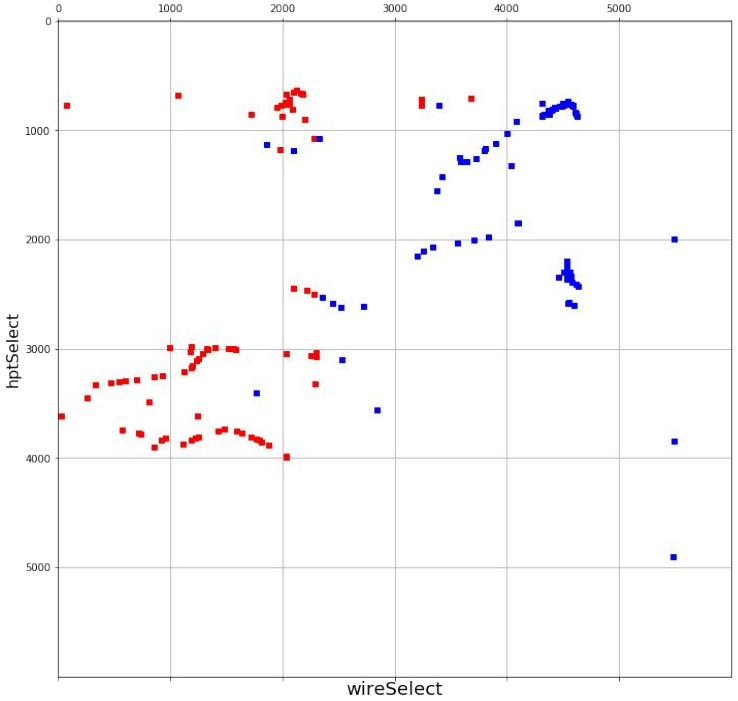
Plot of .root files

./cosmicsintime_redux_15000.root</</</< plane = 2 , tpc = 0 (red) and plane = 1, tpc = 1 (blue)

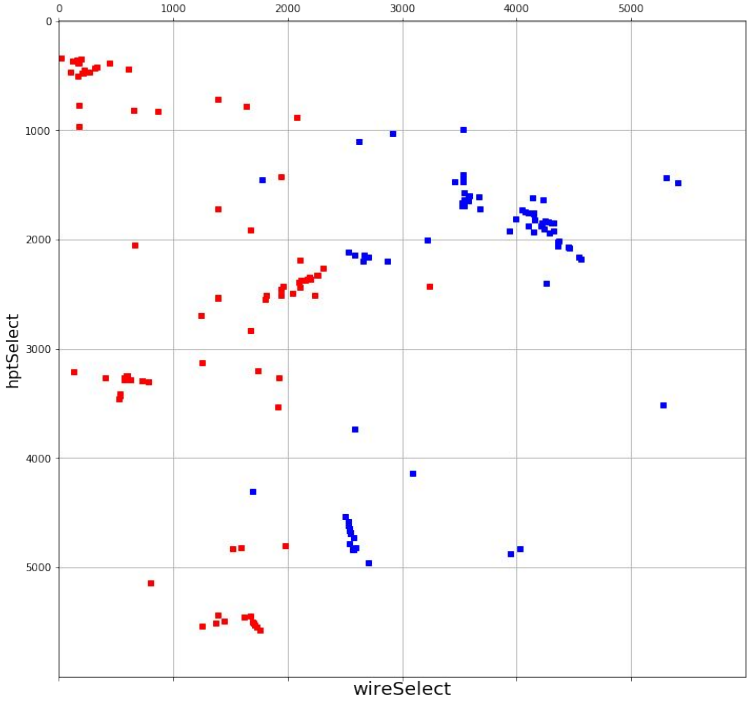


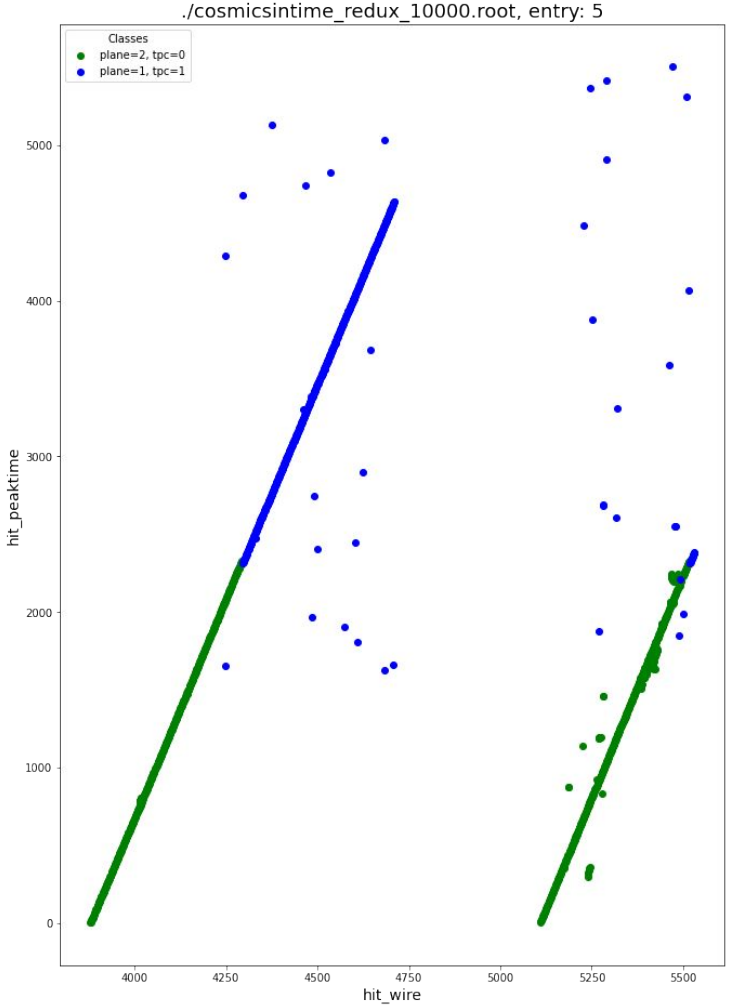
Plot of .root files

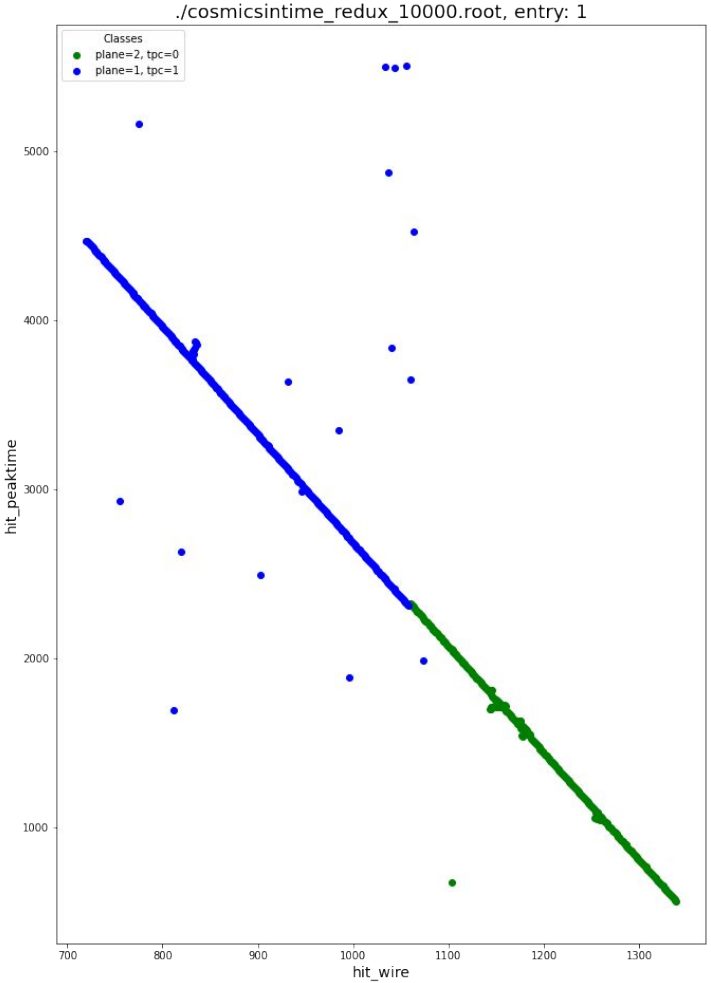
./nue_1300.root</</</< plane = 2 , tpc = 0 (red) and plane = 1, tpc = 1 (blue)



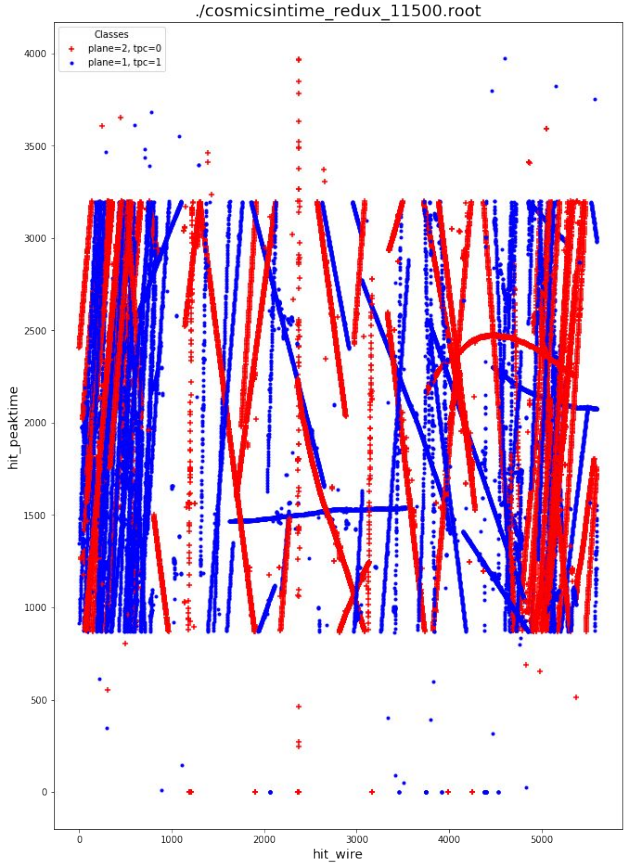
./nue_1000.root</</</< plane = 2 , tpc = 0 (red) and plane = 1, tpc = 1 (blue)

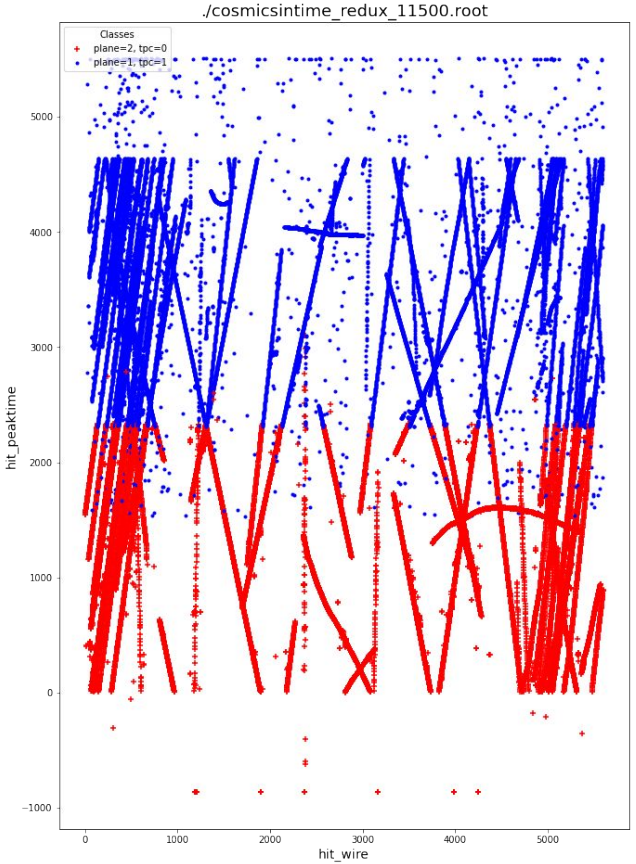



./cosmicsintime_redux_10000.root, entry: 5
Classes
plane=2, tpc=0
plane=1, tpc=1
hit_peaktime
hit_wire
0
1000
2000
3000
4000
5000
4000
4250
4500
4750
5000
5250
5500

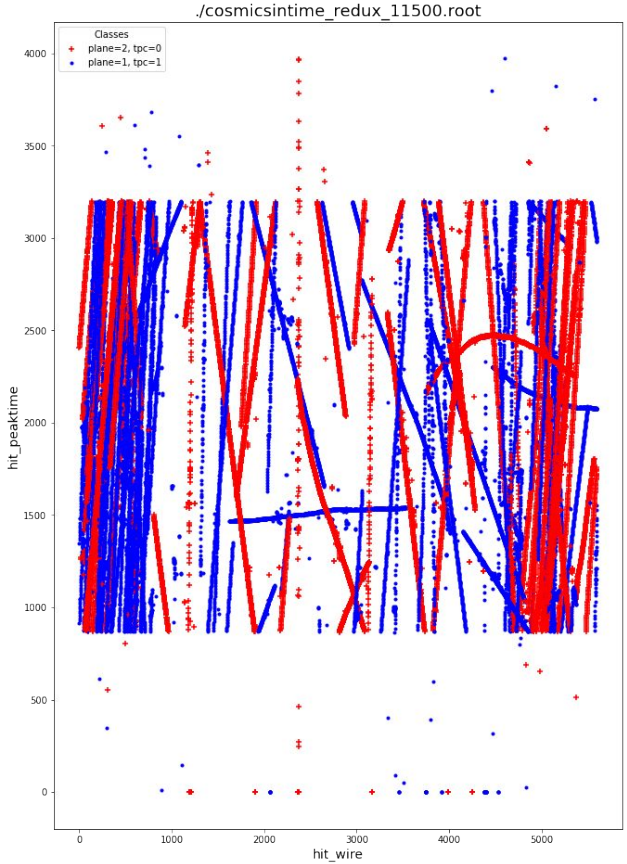

./cosmicsintime_redux_10000.root, entry: 1
Classes
plane=2, tpc=0
plane=1, tpc=1
hit_peaktime
hit_wire
1000
2000
3000
4000
5000
700
800
900
1000
1100
1200
1300

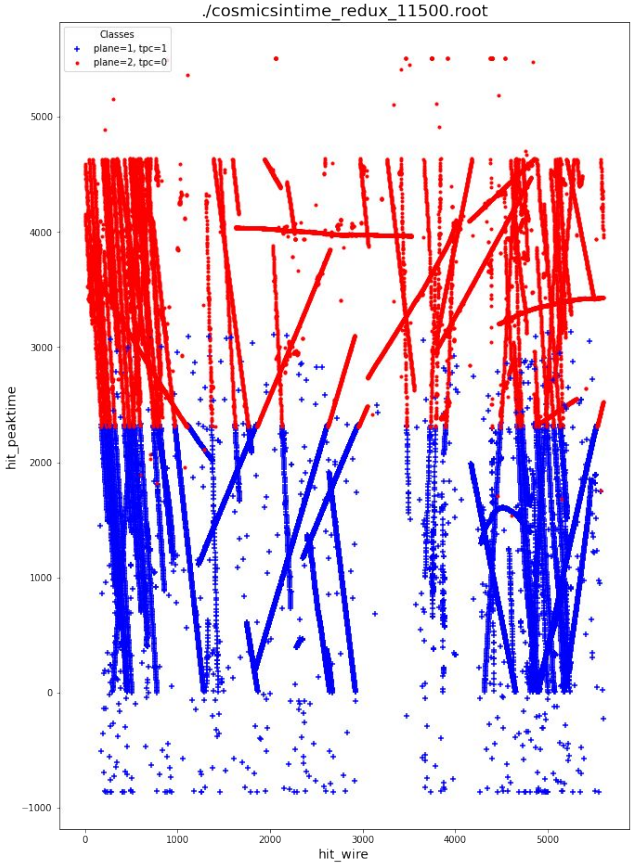
Merging the different TPCs





Merging the different TPCs





Approach to ML

Explore & clean data

- Data contains charged cosmic rays, eNu & noise
- Clean data - remove NA values, same format for all root files
- Plot hit_wire=X vs hit_peak=Y, scatter, histo
- [5728 wires, 4096 samples]

Apply ML algo

- Hit_peak, hit_wire : X
- Event?: Y
- Models & accuracy

➢ RF

Identify & label results

- Result of ML
- Accuracy
- Overfitting - Y/N

Applying ML

Step 1

Split data into train&test

Split data in 70%=train and 30% to test

Shuffle data before inputting

Step 2

Select ML

RandomForest, Seq Vector Machine, Neural network, tSNE

Select appropriate parameters for each algo by trial and error

Step 3

Fit model with data

X = hit_wire, hit_peak

Y = (labels:0,1)

Solution

Selecting proper ML

Select model based on accuracy and proper fit of data to ML type

Not to overfit data

Less time for processing data and building ML model

Implementation(Random forest)

Identify labels & results

Step 1

Size of .roots

Tpc = 0,1, hit_plane = 1 & 2

1 .root = 100 entries,

1 entry = 1000^{+} subentries

To separate diff group of hits to match respective events

5728 wires, 4096 samples

Step 2

Labels used

0 = Cosmic Rays

1 = electron neutrinos

Step 3

Accuracy & overfitting

ROC curve

Predict_accuracy

X: hit_wire, hit_peak

1 event = 5000 wire, 4000 ticks

Make file transformation for 0,1 to plot 2D scatter plot

Fit data to ML

Label Cosmic rays and electron neutrino

Clean data

assign 1s, 0s to prepare for 2D plot

Select RF parameters
Scatter plot = input to ML
Histogram = each event

Both cosmic and nue data are identified by entry and subentry. entry represents the single event and subentry represent single hit. Obviously the ML model should subtract background (cosmic) from signal(nue) for each event. Therefore a set of entry are meaningful for neutrino study. However, machine does not need to know physics! Hereupon, we feed the model a data frame that each subentry stands for a sample. By doing so, the size of sample (training data set) becomes 1000 times bigger. Another matter referring to Icarus TPC. As known active volume of icarus placed at distinctive of icarus tank. Also the high resolution of TPC is defined which called "collection". Also the geometry of ICARUS T600 causes the mirroring effect for printout data that should be take care. According to what was mentioned, the model should train by three set of input data. First, the hits of induction 2 and tpc = 1, then hits of collection with hit_tpc = 0 and lastly the hits that found in mix of these two areas. After finding the best model and evaluating its performance, it would be the time to measure the model over entry and physical event.

Training data for hit_palne = 2 & hit_tpc = 0

			no_hits	hit_wire	hit_peakT
	entry	subentry			
./nue_1700.root	**54**	**669**	760	2726	1499.659424
./cosmicsintime_redux_10000.root	**402**	**1296**	5097	2830	1135.870483
./cosmicsintime_redux_16250.root	**231**	**948**	2007	275	1346.011719
./cosmicsintime_redux_17500.root	**82**	**2737**	4345	1300	2570.357422
./cosmicsintime_redux_19750.root	**101**	**1077**	1267	2978	1872.296631
...	**...**	**...**	...	...	...
./cosmicsintime_redux_18000.root	**100**	**789**	924	5129	1879.041260
	124	**375**	1863	1277	3041.678711
./cosmicsintime_redux_18250.root	**162**	**2870**	3000	5470	1619.719849
./cosmicsintime_redux_19500.root	**88**	**551**	2380	3282	3145.092773
./cosmicsintime_redux_14250.root	**224**	**983**	1490	3013	917.094727

2142828 rows × 3 columns

Training data for hit_palne = 1 & hit_tpc = 1

			no_hits	hit_wire	hit_peakT
	entry	subentry			
./cosmicsintime_redux_18250.root	24	7622	12034	3675	3713.0
./cosmicsintime_redux_12250.root	243	232	340	169	1131.0
./cosmicsintime_redux_15500.root	148	1409	1982	758	2937.0
./cosmicsintime_redux_16000.root	10	4690	6779	4492	1334.0
./cosmicsintime_redux_15000.root	3	665	915	5069	1535.0
...	...	...	...	...	...
./cosmicsintime_redux_17500.root	165	1881	3068	1388	1825.0
	180	1637	2166	3988	1842.0
./cosmicsintime_redux_17750.root	226	1307	2186	1068	1667.0
./cosmicsintime_redux_19000.root	100	312	1124	4356	493.0
./cosmicsintime_redux_14000.root	236	881	3466	1038	935.0

2233122 rows × 3 columns

Training data for mix tpc areas

			no_hits	hit_wire	hit_peakT
	entry	subentry			
./cosmicsintime_redux_11500.root	**154**	**1258**	4548	992	944.124512
./cosmicsintime_redux_15750.root	**34**	**3473**	3924	4940	1071.142700
./cosmicsintime_redux_15250.root	**230**	**734**	988	1662	492.056274
./cosmicsintime_redux_19750.root	**34**	**1358**	2891	2985	4325.000000
./cosmicsintime_redux_19000.root	**2**	**2138**	3095	3433	3865.000000
...	**...**	**...**	...	...	...
./cosmicsintime_redux_11000.root	**26**	**1737**	2038	3551	2735.000000
./cosmicsintime_redux_16500.root	**71**	**963**	1915	262	1006.260254
./cosmicsintime_redux_18250.root	**192**	**1195**	2274	638	1378.065430
./cosmicsintime_redux_14250.root	**229**	**735**	743	5363	1988.508545
./cosmicsintime_redux_11750.root	**248**	**1939**	2388	3542	1131.959106

4375950 rows × 3 columns

Searching for THE BEST RANDOM FOREST MODEL

Parameter selection: time per iteration and total RF model build time

```
pararafo_4parameters = {'bootstrap': [True, False],'max_features': ['auto', 'sqrt'],
             'min_samples_leaf': [1, 10, 40, 80],  'min_samples_split': [2, 10, 35, 60 ],
             'n_estimators': [20, 40, 60, 80]}
```

```
100%|██████████| 256/256 [01:00<00:00,  4.26it/s]
```

```
pararafo = {'bootstrap': [True, False],'max_features': ['auto', 'sqrt'],
             'min_samples_leaf': [1, 10, 40],  'min_samples_split': [2, 10, 35 ],
             'n_estimators': [20, 40, 60]}
```

```
100%|██████████| 108/108 [00:20<00:00,  5.73it/s]
```

```
pararafo = {'bootstrap': [True, False],'max_features': ['auto', 'sqrt'],
             'min_samples_leaf': [1, 10],  'min_samples_split': [2, 10 ],
             'n_estimators': [20, 40]}
```

```
100%|██████████| 32/32 [00:04<00:00,  8.72it/s]
```

- n_estimators = number of trees in the foreset
- max_features = max number of features considered for splitting a node
- max_depth = max number of levels in each decision tree
- min_samples_split = min number of data points placed in a node before the node is split
- min_samples_leaf = min number of data points allowed in a leaf node
- bootstrap = method for sampling data points (with or without replacement)

Searching for THE BEST RANDOM FOREST MODEL

```
pararafo = {'bootstrap': [True, False],'max_features': ['auto', 'sqrt'],
            'min_samples_leaf': [1],  'min_samples_split': [10 ],
            'n_estimators': [37,47]}
```

```
pararafo
```

```
{'bootstrap': [True, False],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1],
 'min_samples_split': [10],
 'n_estimators': [37, 47]}
```

- n_estimators = number of trees in the forest
- max_features = max number of features considered for splitting a node
- max_depth = max number of levels in each decision tree
- min_samples_split = min number of data points placed in a node before the node is split
- min_samples_leaf = min number of data points allowed in a leaf node
- bootstrap = method for sampling data points (with or without replacement)

Searching for THE BEST RANDOM FOREST MODEL for hit_palne = 2 & hit_tpc = 0

```
best_randomforest.model
```

```
RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=10,
                       min_weight_fraction_leaf=0.0, n_estimators=40,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

[174]
```
best_randomforest.Display_Results()
```

values	0.0100	0.0200	0.0500	0.1000	0.2000
closest_fpr	0.0100	0.0200	0.0500	0.1000	0.2003
tpr	0.9836	0.9896	0.9952	0.9978	0.9995
threshold	0.2612	0.1952	0.1135	0.0554	0.0063

Searching for THE BEST RANDOM FOREST MODEL for mix tpcs

In [72]:
```
pararafo = {'bootstrap': [True, False],'max_features': ['auto', 'sqrt'],
            'min_samples_leaf': [1],  'min_samples_split': [10 ],
            'n_estimators': [39, 44]}
```

In [65]:
```
best_randomforest.model
```

```
RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=10,
                       min_weight_fraction_leaf=0.0, n_estimators=42,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

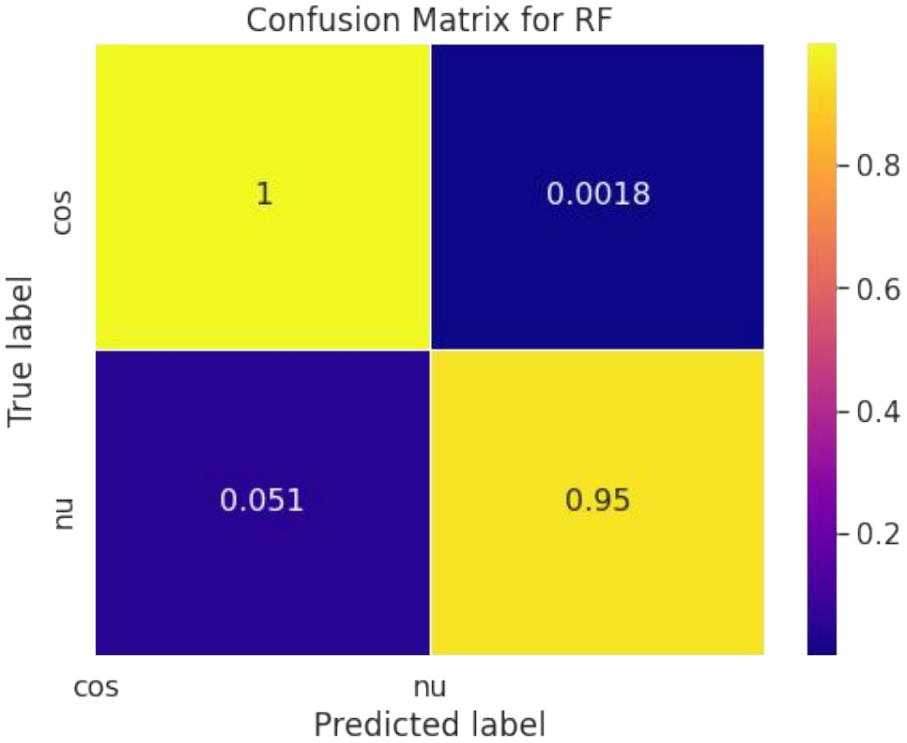
In [161]:
```
Y_pred_test = rf.predict(X_test)
accuracy_score(Y_test, Y_pred_test)
```

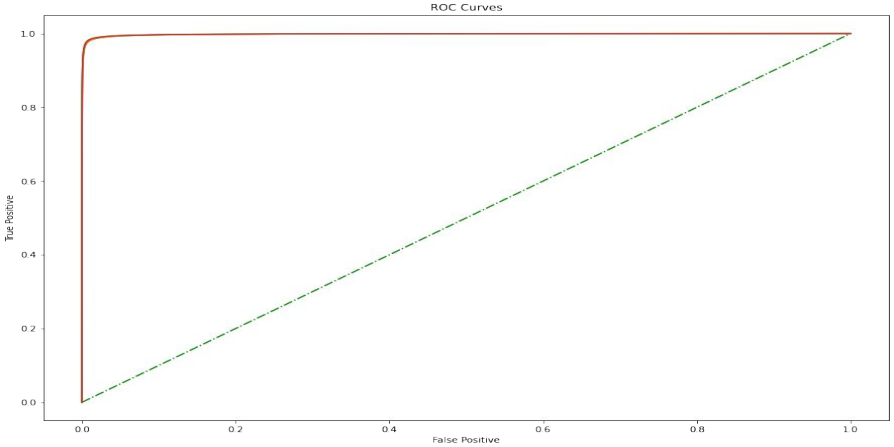
```
0.9935416690815376
```

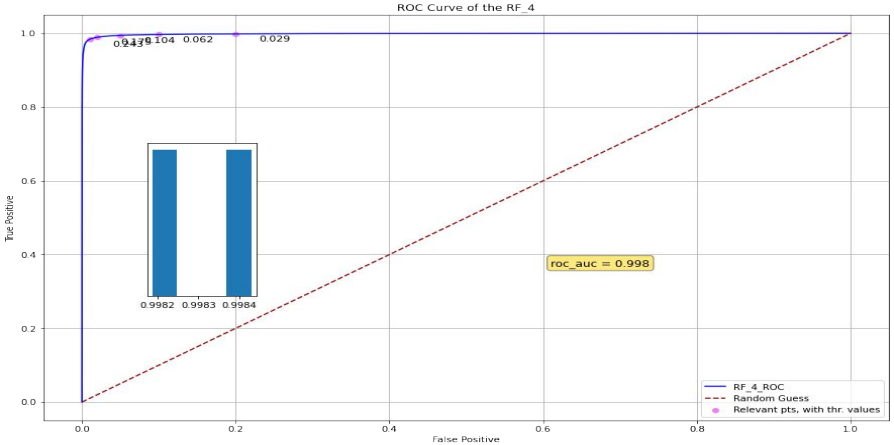
Results and evaluation the best RF model for input data of compound tpc areas

Results for input data of compound tpc areas

AUC ROC







```
1 best_randomforest.model
```

```
RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=10,
                       min_weight_fraction_leaf=0.0, n_estimators=42,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Score of the best model ¶

```
1 Y_pred_test = rf.predict(X_test)
2 accuracy_score(Y_test, Y_pred_test)
```

```
0.9910309802301305
```

```
wall time of building RFs:  2668.222023963928 sec
wall time of building RFs:  44.47036706606547 min
```

Confusion Matrix

```
1 con_mat = confusion_matrix(Y_test, Y_pred_test)
2 print(con_mat)
```

```
[[932407   1662]
 [  8150 151769]]
```

```
1 best_randomforest.Display_Results()
```

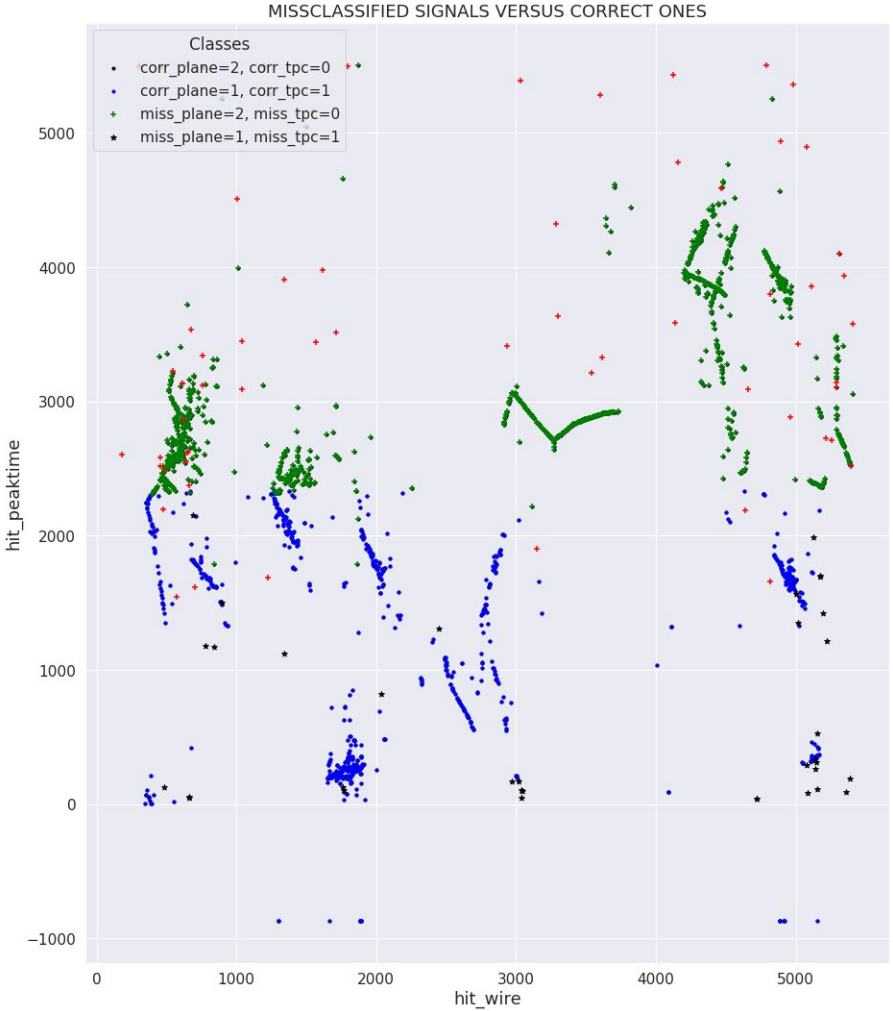
values	0.0100	0.0200	0.0500	0.1000	0.2000
False Positive	0.0100	0.0200	0.0500	0.1000	0.2000
True Positive	0.9836	0.9891	0.9945	0.9968	0.9984
threshold	0.2432	0.1748	0.1038	0.0624	0.0292

```
true negative(TN):=> 932407 || false positive(FP) :=> 1662
================================================================
false negative(FN):=> 8150  || true positive(TP) :=> 151769
----------------------------------------------------------------
P => 159919 N => 934069
----------------------------------------------------------------
Sensitivity or true positive rate(TPR) ==> 0.9490366998292886
MISS RATE or false negative rate(FNR) ==> 0.05096330017071138
________________________________________________________________
specificity, selectivity or true negative rate(TNR) ==> 0.9982206881932705
fall-out or false positive rate (FPR) ==> 0.0017793118067295222
________________________________________________________________
precision or positive predictive value(PPV) ==> 0.9891677692252544
false discovery rate (FDR) ==> 0.0108322307747456
________________________________________________________________
negative predictive value(NPV) ==> 0.9913349217538119
false omission rate (FOR) ==> 0.008665078246188118
________________________________________________________________
threat score(TS) or critical success index(CSI) ==> 0.9392750385255693
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\//\/\/\/\/\/\//\/\/\//\/\/\/\/\//
prevalence threshold(PT) ==> 0.041502614762624054
```

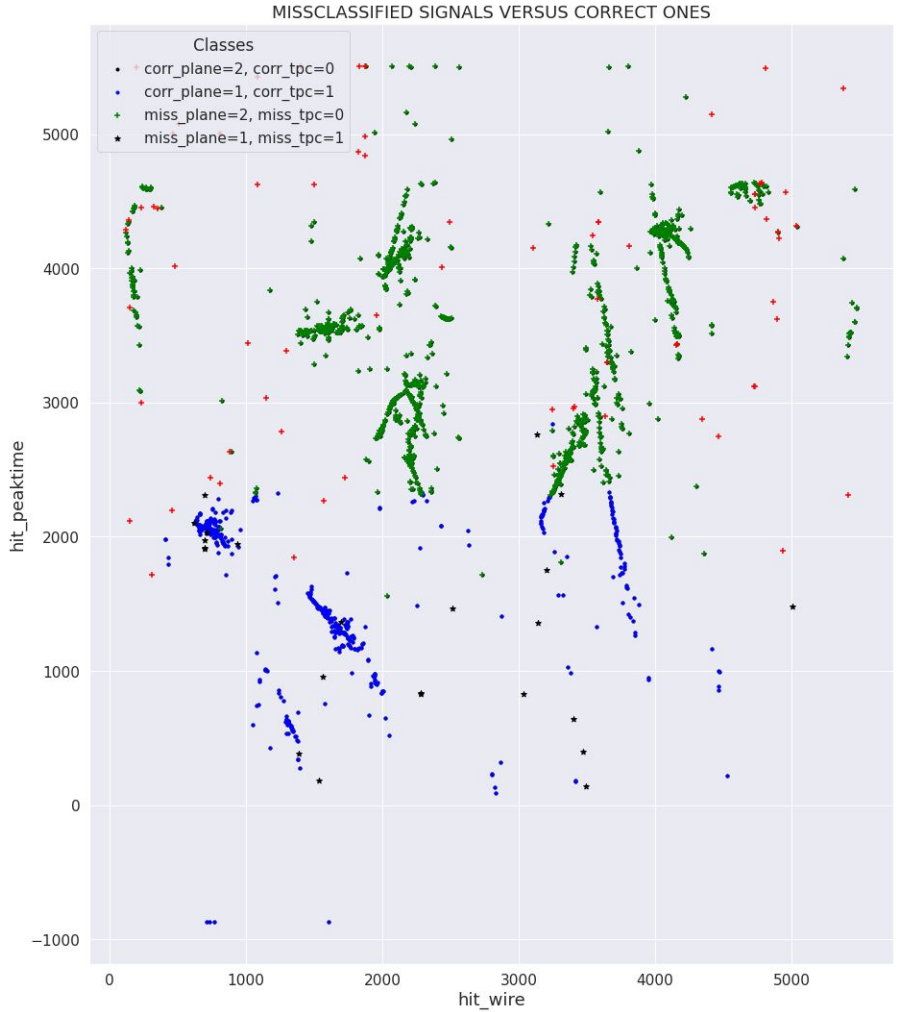
Misclassified nues

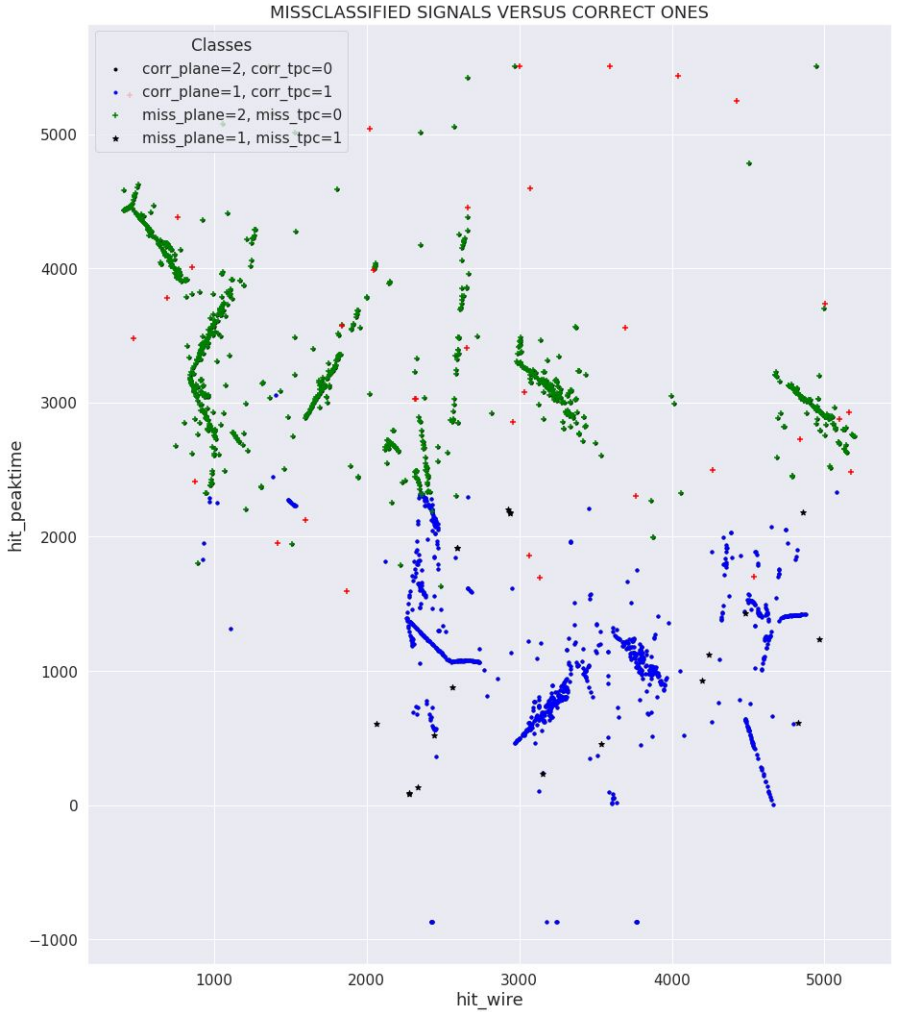
	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
./nue_0.root	1	431	0	2	1369	920	2280.197754	1	ev
		819	1	1	1369	849	2304.000000	1	ev
	2	400	1	1	954	4731	4551.000000	1	ev
		402	1	1	954	4732	4551.000000	1	ev
		403	1	1	954	4732	4450.000000	1	ev
...	...	...	...	...	...	...	...	...	...
./nue_900.root	98	1310	1	1	2229	5317	2005.000000	1	ev
		1312	1	1	2229	5322	3190.000000	1	ev
		1313	1	1	2229	5323	3185.000000	1	ev
	99	327	0	2	333	376	21.757507	1	ev
		329	0	2	333	398	4.550354	1	ev

8150 rows × 7 columns



The performance of the best RF model to discriminate nues:
The hit of induction and collection in the same canvas.


MISSCLASSIFIED SIGNALS VERSUS CORRECT ONES
Classes
corr_plane=2, corr_tpc=0
corr_plane=1, corr_tpc=1
miss_plane=2, miss_tpc=0
miss_plane=1, miss_tpc=1
hit_peaktime
hit_wire
5000
4000
3000
2000
1000
0
−1000
0
1000
2000
3000
4000
5000


MISSCLASSIFIED SIGNALS VERSUS CORRECT ONES
Classes
corr_plane=2, corr_tpc=0
corr_plane=1, corr_tpc=1
miss_plane=2, miss_tpc=0
miss_plane=1, miss_tpc=1
hit_peaktime
hit_wire
5000
4000
3000
2000
1000
0
−1000
1000
2000
3000
4000
5000

Misclassified cosmics

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
	17	**917**	1	1	1721	2463	3732.0	0	cos
		756	1	1	1389	4065	4835.0	0	cos
./cosmicsintime_redux_10000.root	**25**	**793**	1	1	1389	4128	4260.0	0	cos
		811	1	1	1389	4142	4699.0	0	cos
	29	**1196**	1	1	2532	3083	1537.0	0	cos
...	**...**	**...**	...	...	...	...	...	...	...
		1180	1	1	1219	2305	2630.0	0	cos
		1188	1	1	1219	2483	5505.0	0	cos
./cosmicsintime_redux_19750.root	**246**	**1194**	1	1	1219	2560	1820.0	0	cos
		1195	1	1	1219	2588	1743.0	0	cos
		1196	1	1	1219	2591	2766.0	0	cos

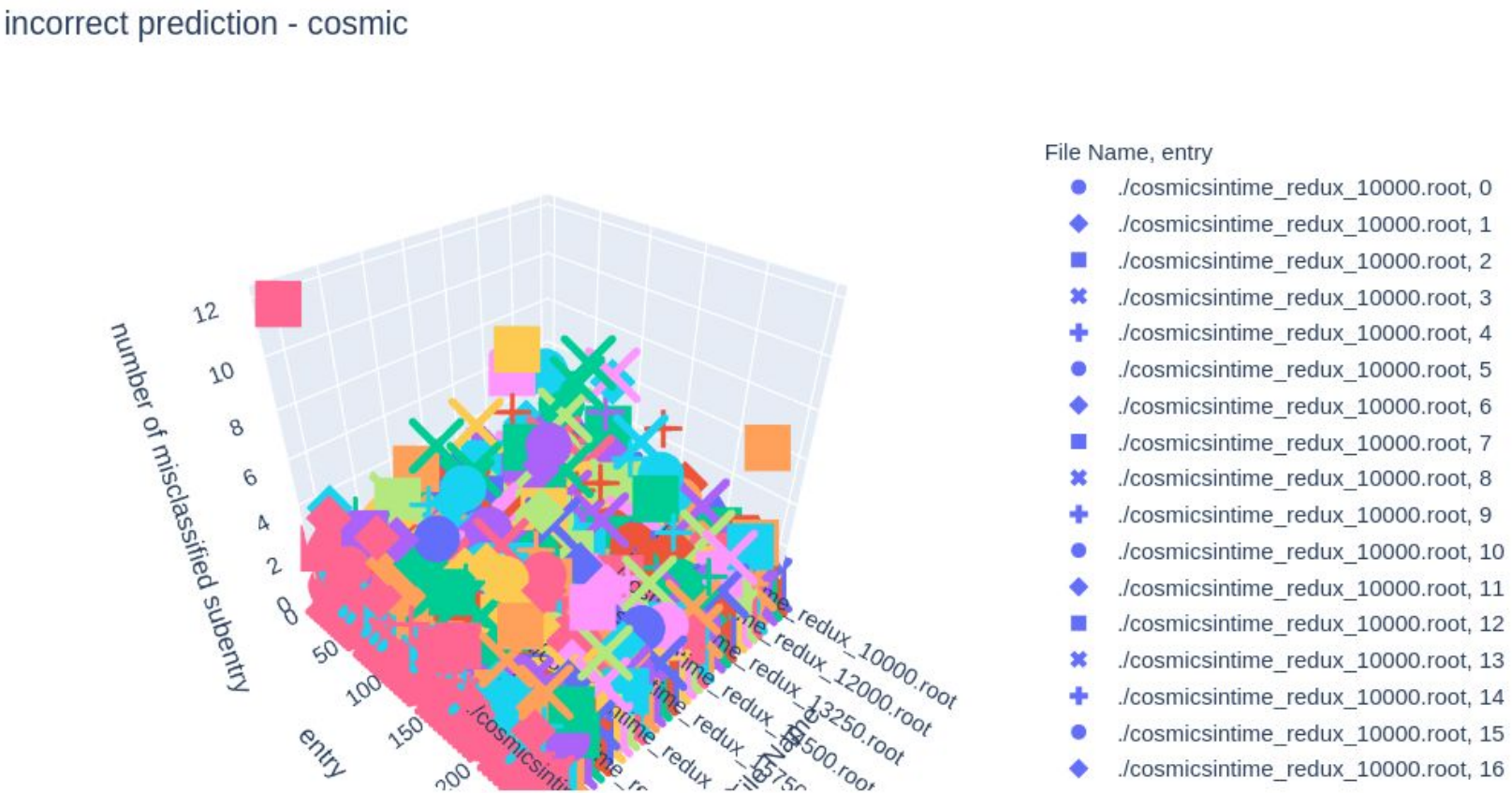
1662 rows × 7 columns

❖ Table of Frequency of correct predicted cosmics Vs.Misclassified backgrounds that were predicted as signals

	fName	entry	no_corr_subentry	no_miss_subentry
0	./cosmicsintime_redux_10000.root	0	20	0
1	./cosmicsintime_redux_10000.root	1	118	0
2	./cosmicsintime_redux_10000.root	2	181	0
3	./cosmicsintime_redux_10000.root	3	153	0
4	./cosmicsintime_redux_10000.root	4	46	0
...	...	...	...	...
9245	./cosmicsintime_redux_19750.root	245	68	0
9246	./cosmicsintime_redux_19750.root	246	79	8
9247	./cosmicsintime_redux_19750.root	247	39	0
9248	./cosmicsintime_redux_19750.root	248	56	0
9249	./cosmicsintime_redux_19750.root	249	81	0

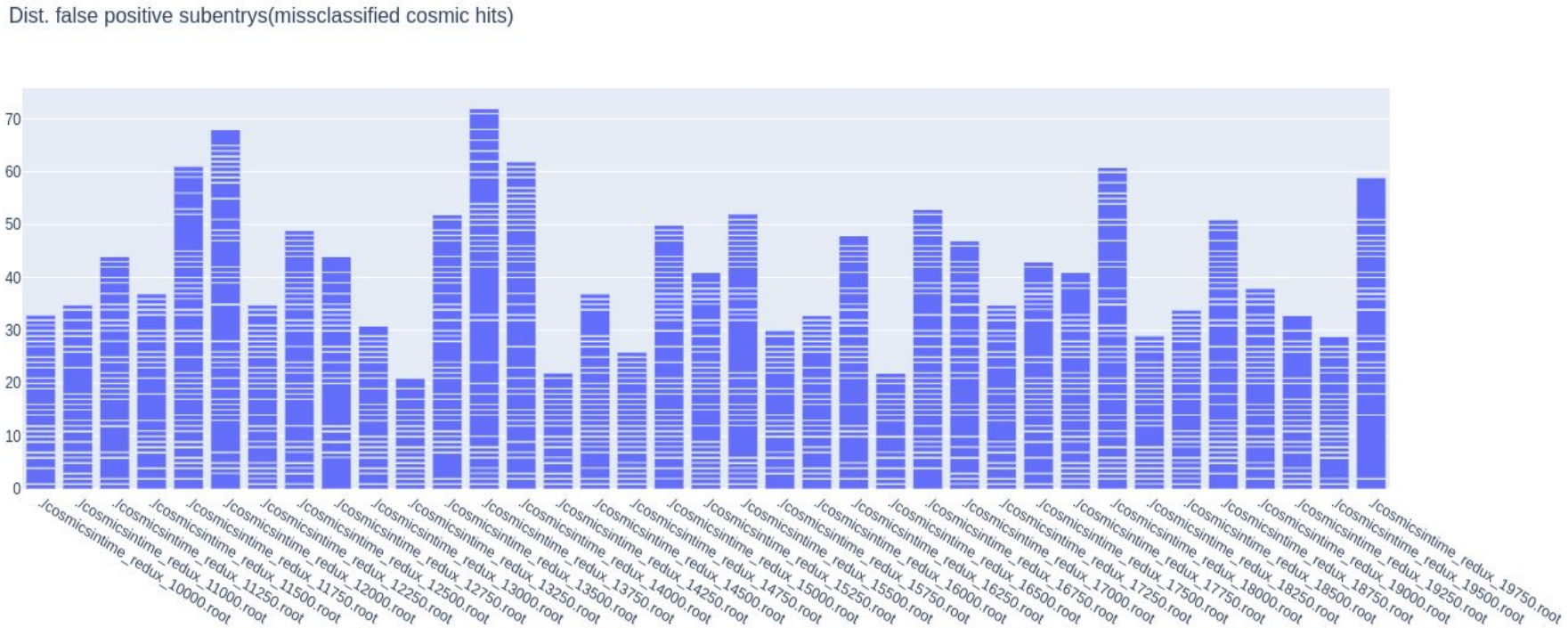
9250 rows × 4 columns

Distribution of false subentry of each entry for misclassified cosmic hits

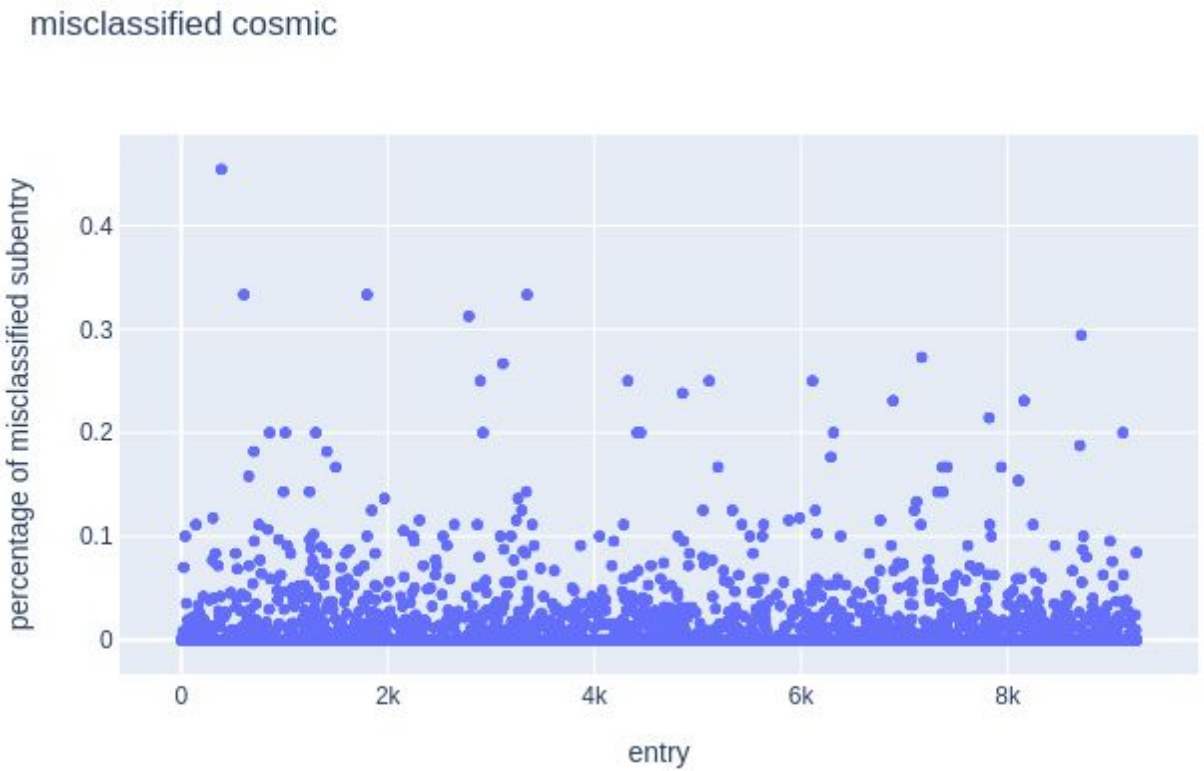


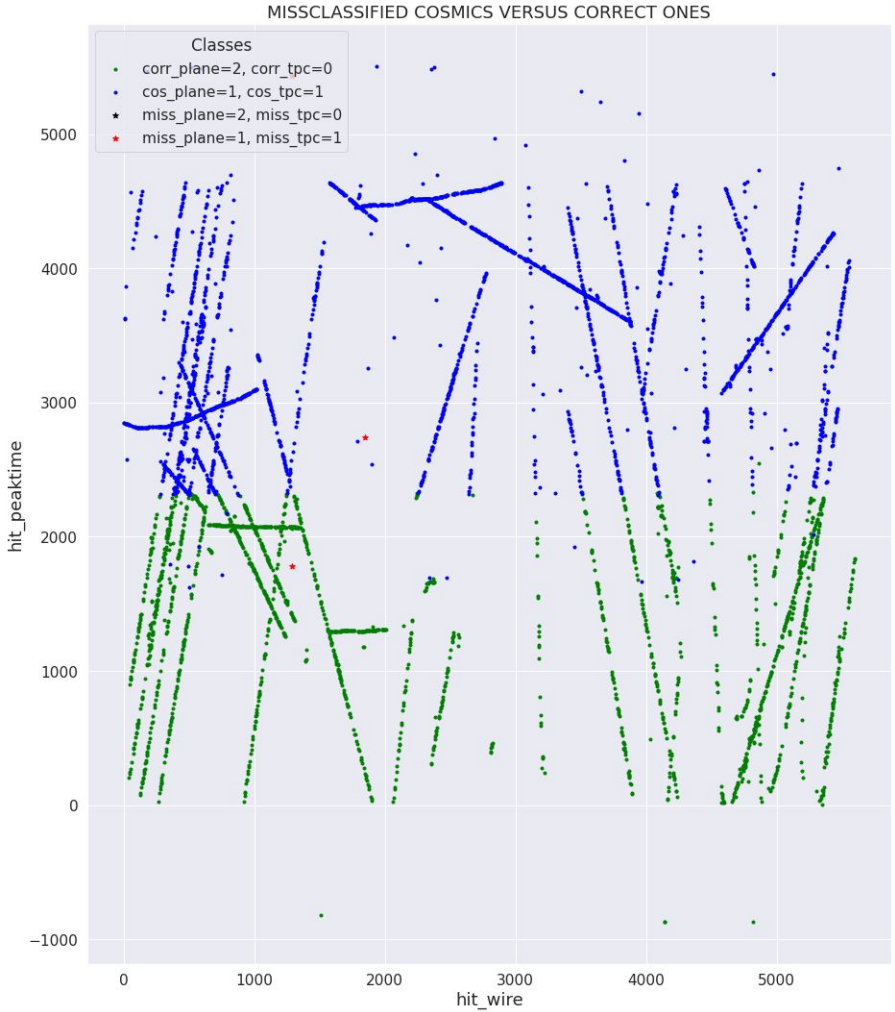
https://colab.research.google.com/drive/1W2XZqT5-7CSIKPwEqsKi0av7PWMJRaQi?usp=sharing

❖ Plot of Frequency of Misclassified backgrounds that are predicted as signals

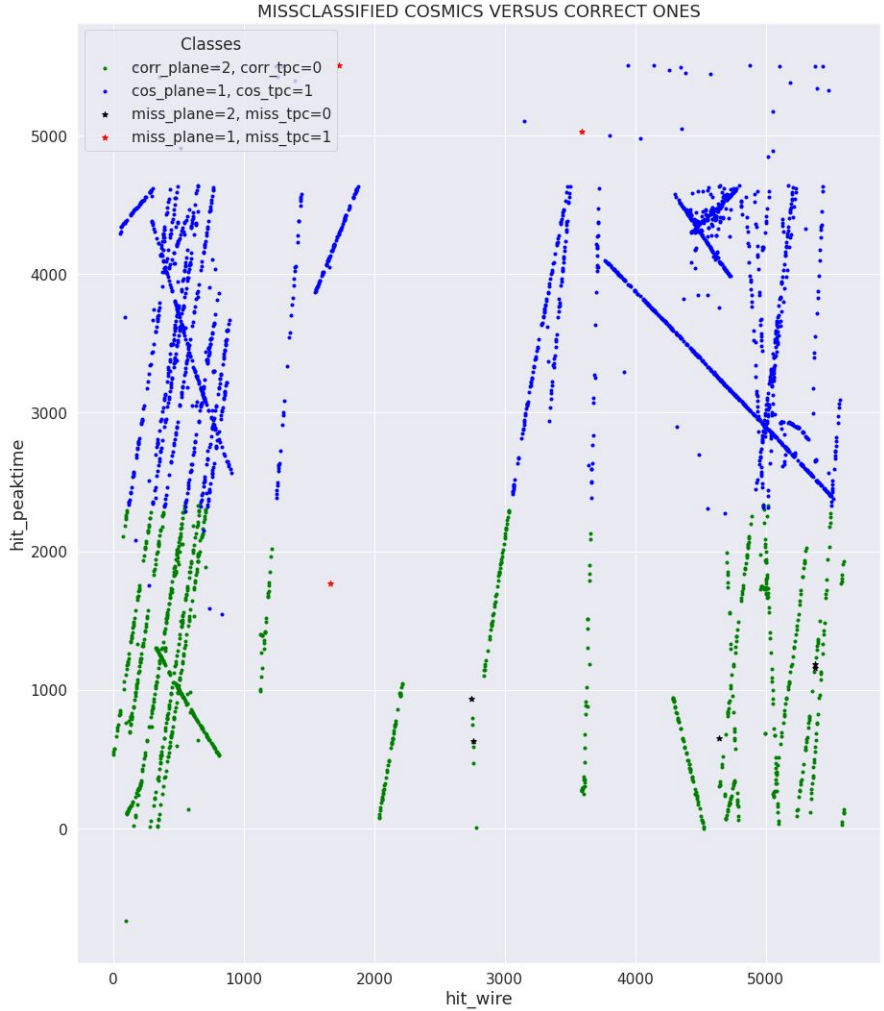


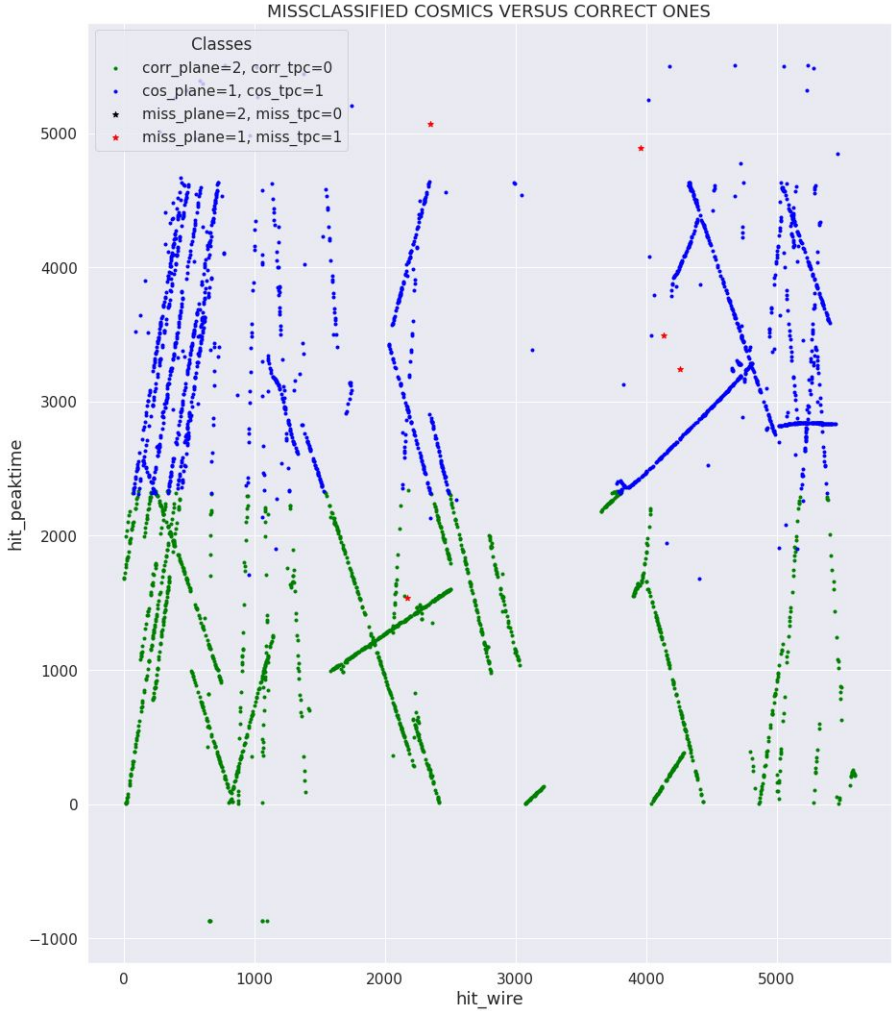
❖ Plot : Percentage of Misclassified subentry(single hits) for each entry - cosmic





The performance of the best RF model to discriminate cosmics:
The hit of induction and collection in the same canvas.


MISSCLASSIFIED COSMICS VERSUS CORRECT ONES
Classes
corr_plane=2, corr_tpc=0
cos_plane=1, cos_tpc=1
miss_plane=2, miss_tpc=0
miss_plane=1, miss_tpc=1
hit_peaktime
hit_wire
5000
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000

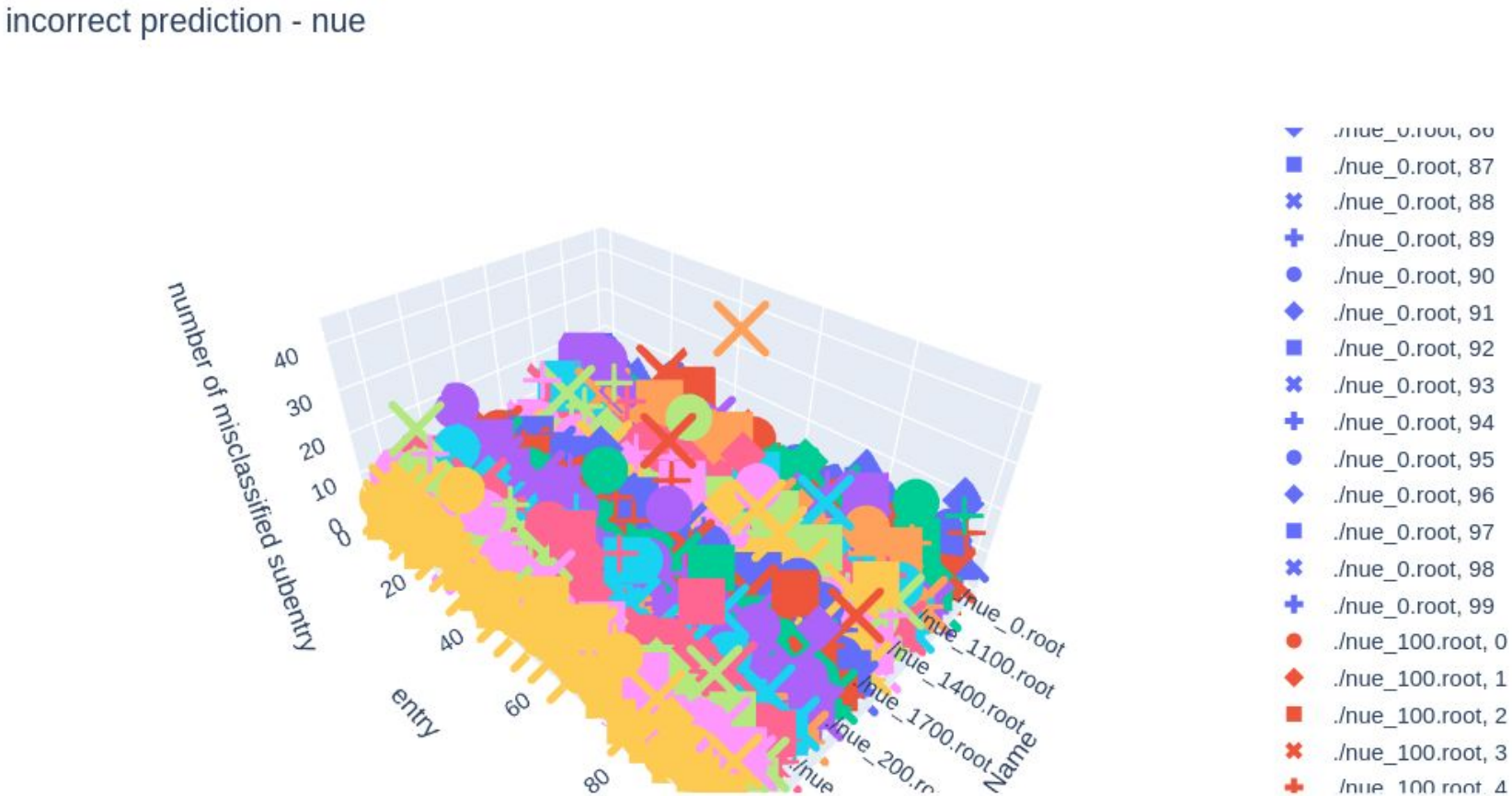

MISSCLASSIFIED COSMICS VERSUS CORRECT ONES
Classes
corr_plane=2, corr_tpc=0
cos_plane=1, cos_tpc=1
miss_plane=2, miss_tpc=0
miss_plane=1, miss_tpc=1
hit_peaktime
hit_wire
5000
4000
3000
2000
1000
0
−1000
0
1000
2000
3000
4000
5000

❖ Table of Frequency of correct prediction Vs.Misclassified nues

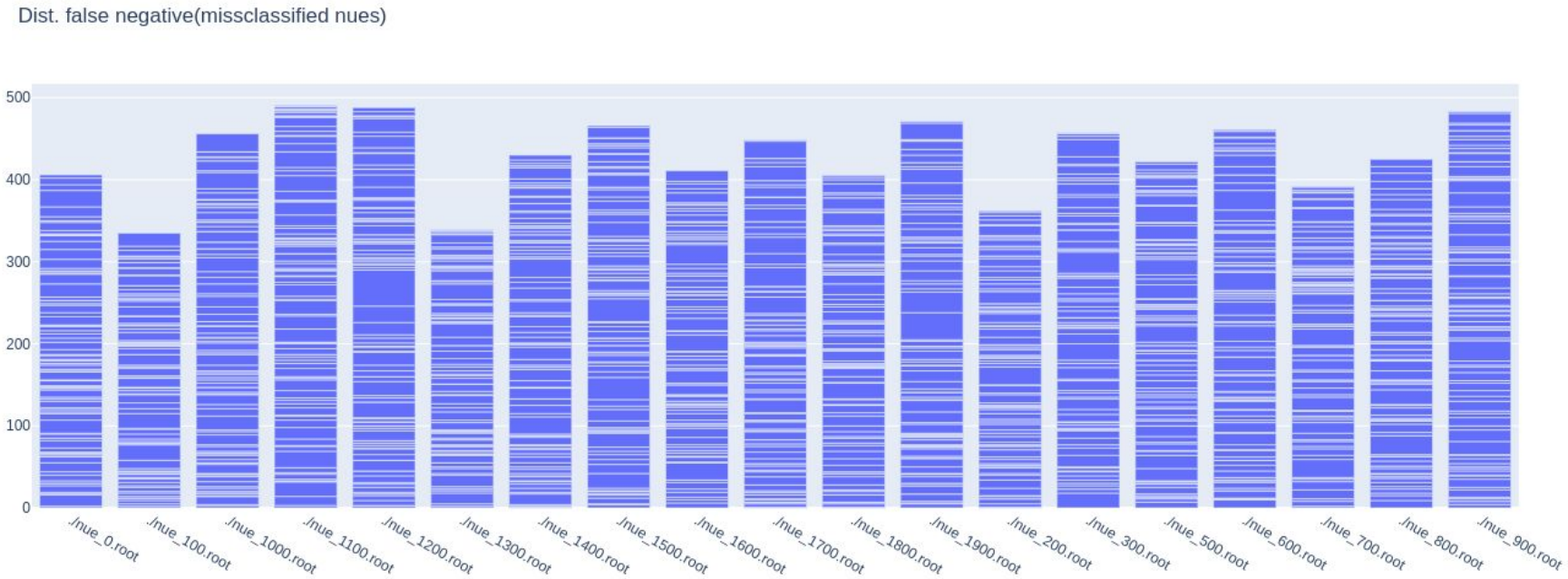
	fName	entry	no_corr_subentry	no_miss_subentry
0	./nue_0.root	0	2	0
1	./nue_0.root	1	97	2
2	./nue_0.root	2	54	14
3	./nue_0.root	3	82	1
4	./nue_0.root	4	13	2
...	...	...	...	...
1895	./nue_900.root	95	261	0
1896	./nue_900.root	96	104	7
1897	./nue_900.root	97	201	2
1898	./nue_900.root	98	140	12
1899	./nue_900.root	99	19	2

1900 rows × 4 columns

Distribution of false subentry of each entry for misclassified nues

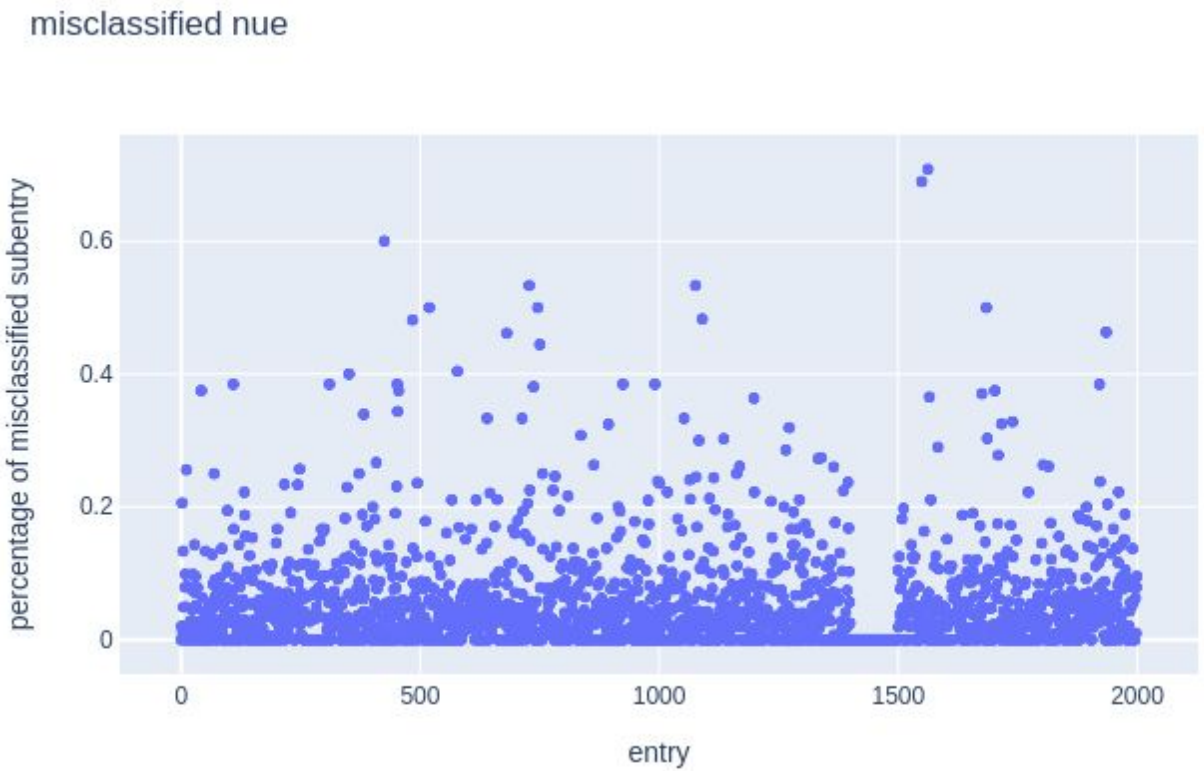


❖ Plot of Frequency of misclassified nues



https://colab.research.google.com/drive/1W2XZqT5-7CSlKPwEgsKi0av7PWMJRaOj#scrollTo=J2e5JZXNFAdW&line=2&uniqifier=1

❖ Plot : Percentage of Misclassified subentry(single hits) for each entry - nue

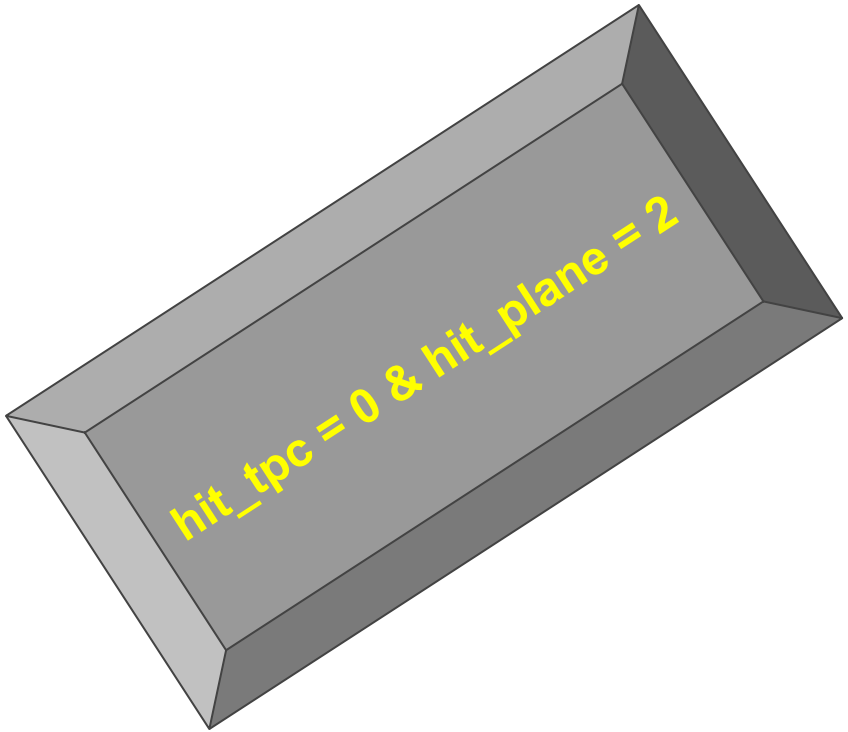


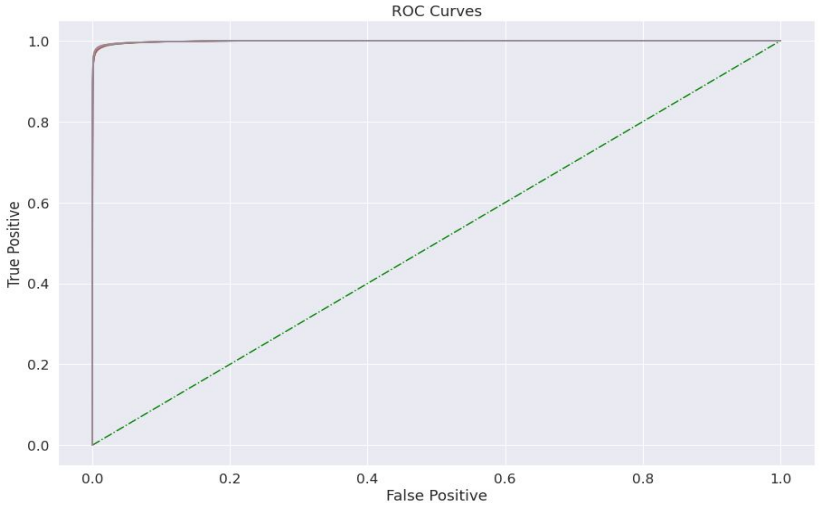
Performance of the model over physical event

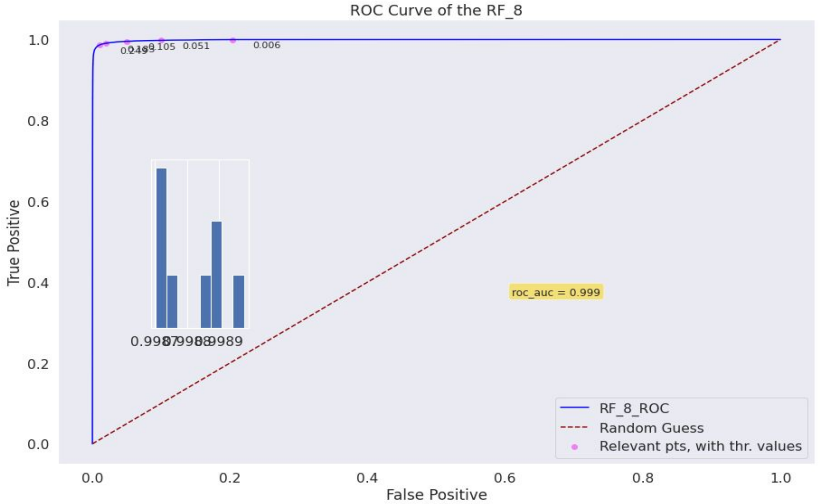
```
true negative(TN):=> 9261 || false positive(FP) :=> 231
=================================================================
false negative(FN):=> 738  || true positive(TP) :=> 1162
-----------------------------------------------------------------
P => 1900 N => 9492
-----------------------------------------------------------------
Sensitivity or true positive rate(TPR) ==> 0.611578947368421
MISS RATE or false negative rate(FNR) ==> 0.388421052631579
________________________________________________________________________________
specificity, selectivity or true negative rate(TNR) ==> 0.9756637168141593
fall-out or false positive rate (FPR) ==> 0.024336283185840690
________________________________________________________________________________
precision or positive predictive value(PPV) ==> 0.8341708542713567
false discovery rate (FDR) ==> 0.16582914572864327
________________________________________________________________________________
negative predictive value(NPV) ==> 0.9261926192619262
false omission rate (FOR) ==> 0.0738073807380738
________________________________________________________________________________
threat score(TS) or critical success index(CSI) ==> 0.5452839042702956
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\//\/\/\/\/\/\//\/\/\//\/\/\/\/\/
prevalence threshold(PT) ==> 0.16630587854259013
================================================================================
================================================================================
Accuracy of the model - entry level(ACC) ==> 0.914940308988764
================================================================================
================================================================================
```

Apart TPC

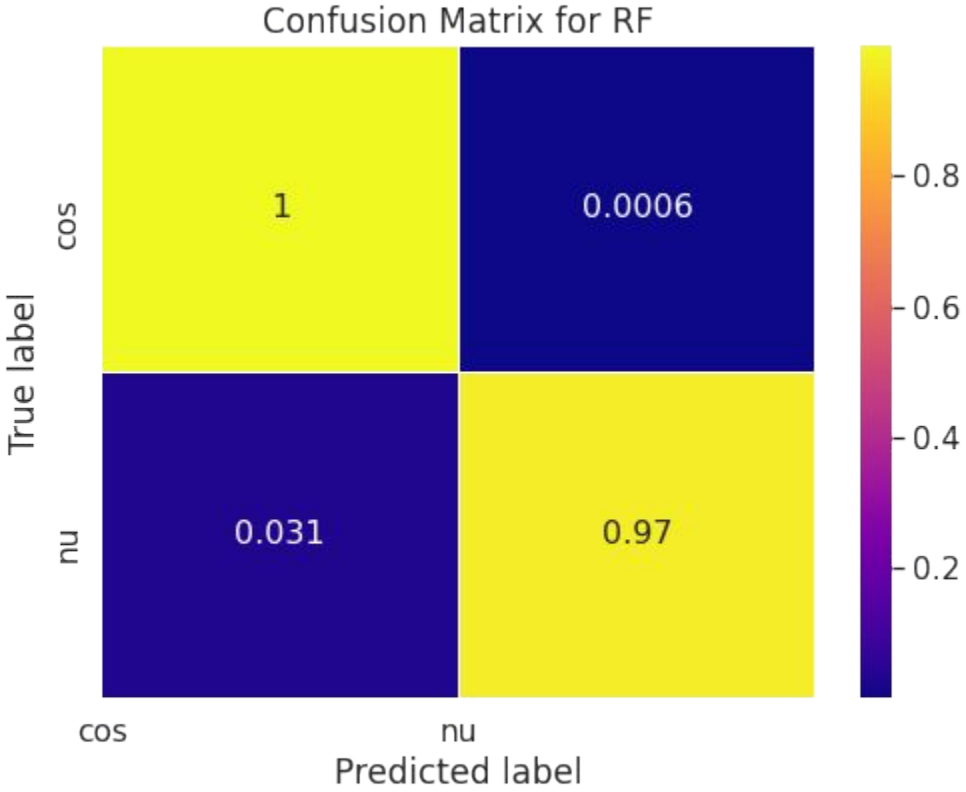
ROC Curves Analysis







Confusion matrix



97% of eNeu = correct label

3.1% eNeu = wrong labeled as cosmic

0.06% cosmic = wrong labeled as eNue

RMS error for RF = 7.088818666394833

Accuracy of Random Forest : 0.9949748649914 97

hit_tpc = 0 & hit_plane = 2

```
con_mat = confusion_matrix(Y_test, Y_pred_test)
print(con_mat)
```

```
[[456839    274]
 [  2418  76176]]
```

```
# mse = mean_squared_error(actual, predicted)
predicted = rf.predict(X_test)
mse = mean_squared_error(Y_test, predicted)
rmse = math.sqrt(mse)
print('root mean square error for Random Forest',100*max(0,rmse))
print('accuracy of random forest',rf.score(X_test,Y_test))
```

```
root mean square error for Random Forest 7.088818666394833
accuracy of random forest 0.9949748649914972
```

hit_tpc = 0 & hit_plane = 2

```
true negative(TN):=> 456839 || false positive(FP) :=> 274
================================================================
false negative(FN):=> 2418  || true positive(TP) :=> 76176
----------------------------------------------------------------
P => 78594 N => 457113
----------------------------------------------------------------
Sensitivity or true positive rate(TPR) ==> 0.9692342926940988
MISS RATE or false negative rate(FNR) ==> 0.030765707305901224
________________________________________________________________
specificity, selectivity or true negative rate(TNR) ==> 0.9994005858507634
fall-out or false positive rate (FPR) ==> 0.0005994141492365879
________________________________________________________________
precision or positive predictive value(PPV) ==> 0.9964159581425769
false discovery rate (FDR) ==> 0.0035840418574231148
________________________________________________________________
negative predictive value(NPV) ==> 0.9947349740994693
false omission rate (FOR) ==> 0.005265025900530684
________________________________________________________________
threat score(TS) or critical success index(CSI) ==> 0.9658670183090734
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\//\/\/\/\/\/\/\//\/\/\/\//\/\/\/\/\/\//
prevalence threshold(PT) ==> 0.024265038013227522
```

hit_tpc = 0 & hit_plane = 2

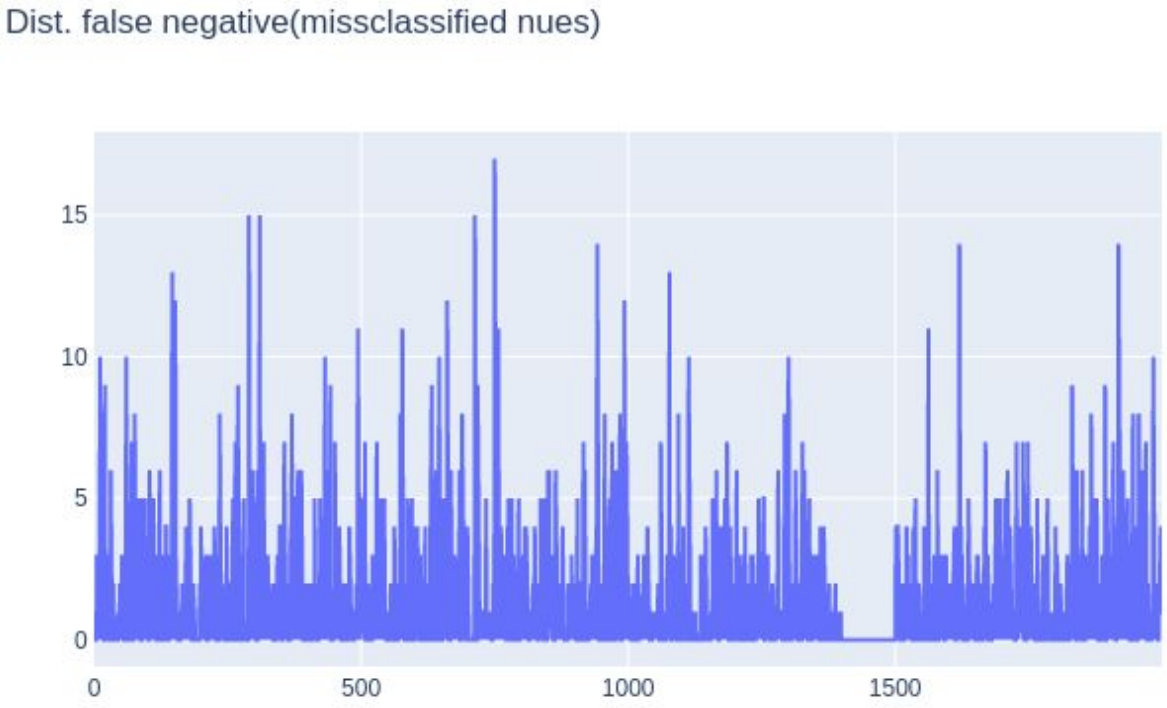
Misclassified nues that were predicted as cosmics; Induction2

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
	1	431	0	2	1369	920	3145.197754	1	ev
		1446	0	2	2343	3335	2734.616455	1	ev
./nue_0.root	5	1447	0	2	2343	3359	2672.789551	1	ev
		2145	0	2	2343	3955	2604.878662	1	ev
	6	1574	0	2	2194	3736	2564.831299	1	ev
...	...	...	...	...	...	...	...	...	...
		437	0	2	3118	4572	2537.922852	1	ev
	97	450	0	2	3118	4785	3023.055420	1	ev
./nue_900.root		583	0	2	3118	4908	2719.838867	1	ev
	98	2224	0	2	2229	5240	932.359863	1	ev
	99	295	0	2	333	347	1064.834595	1	ev

2418 rows × 7 columns

hit_tpc = 0 & hit_plane = 2

Distribution of false subentry of each entry for misclassified nues

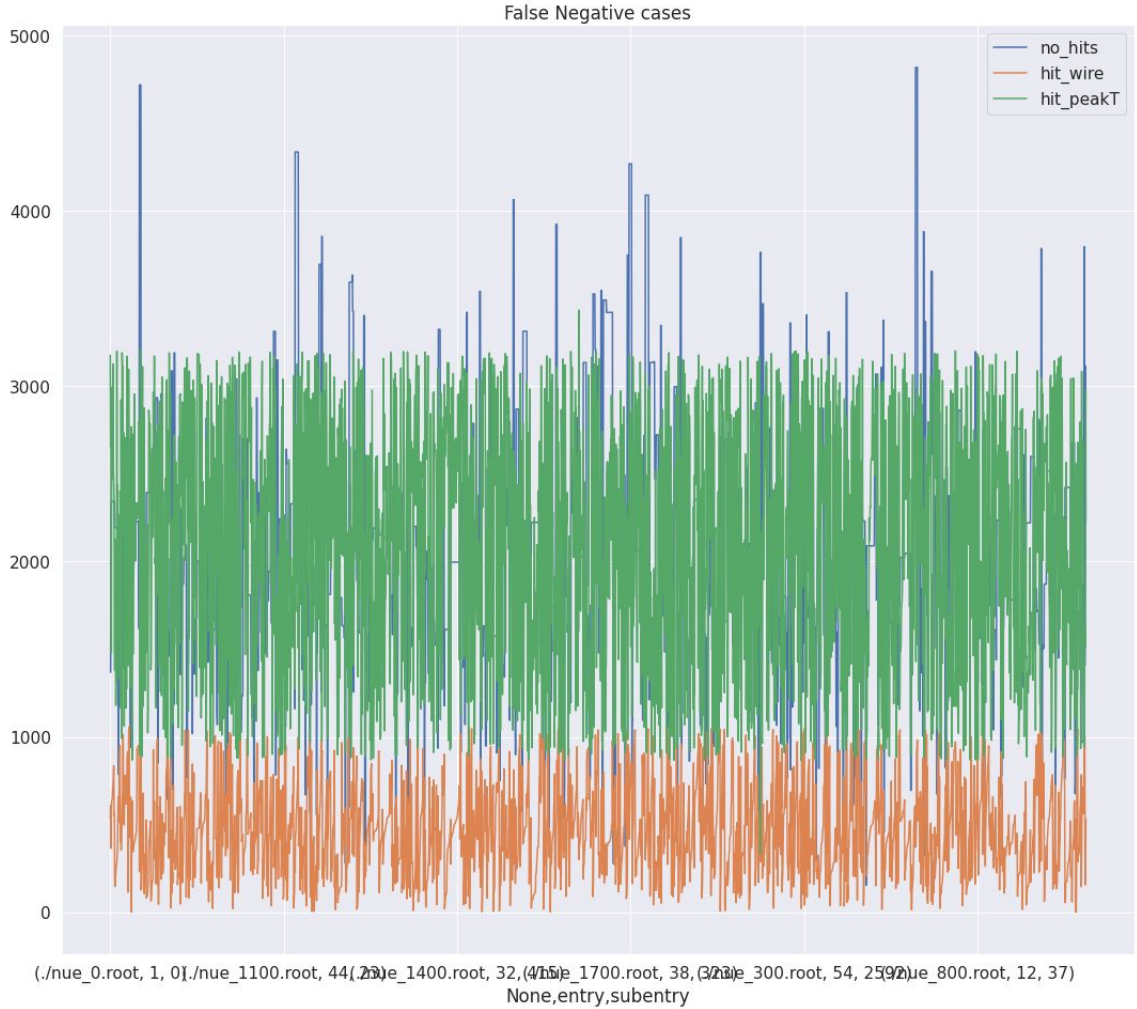


hit_tpc = 0 & hit_plane = 2

Neutrino tracks were predicted as Cosmic by RF

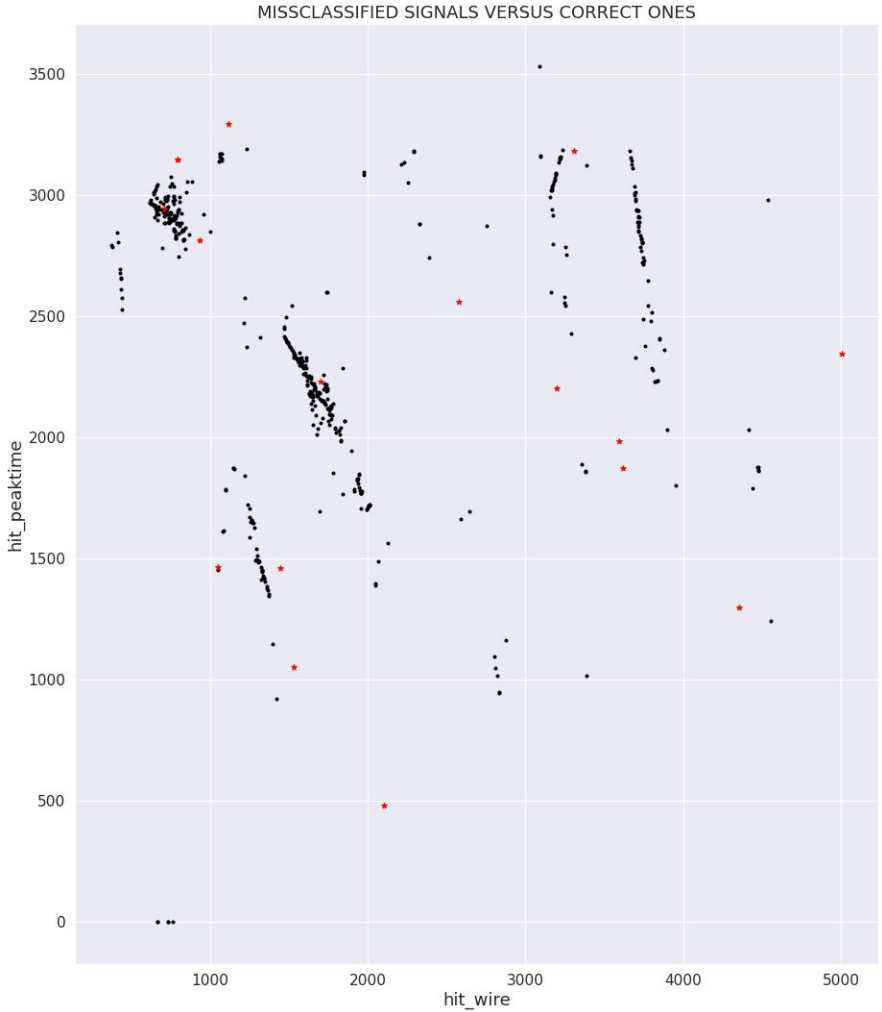
The points with same color belong to same entry

hit_tpc = 0 & hit_plane = 2



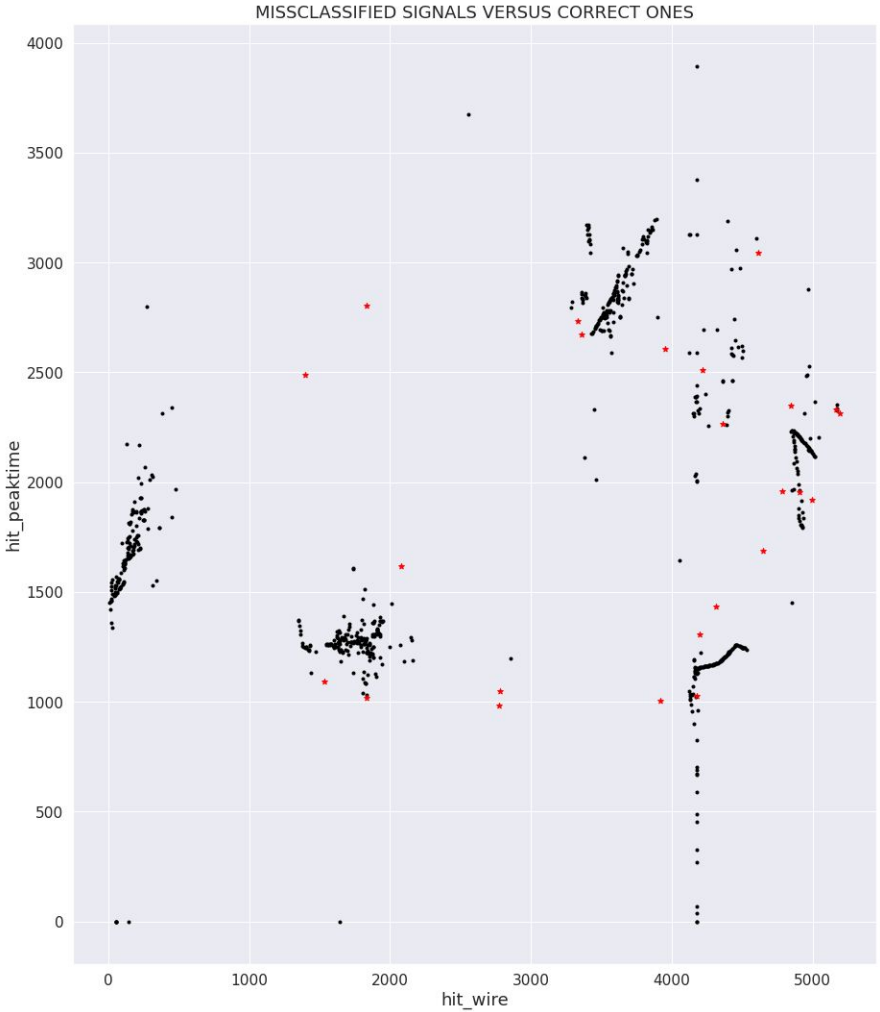
Neutrino tracks predicted as Cosmic by RF

hit_tpc = 0 & hit_plane = 2

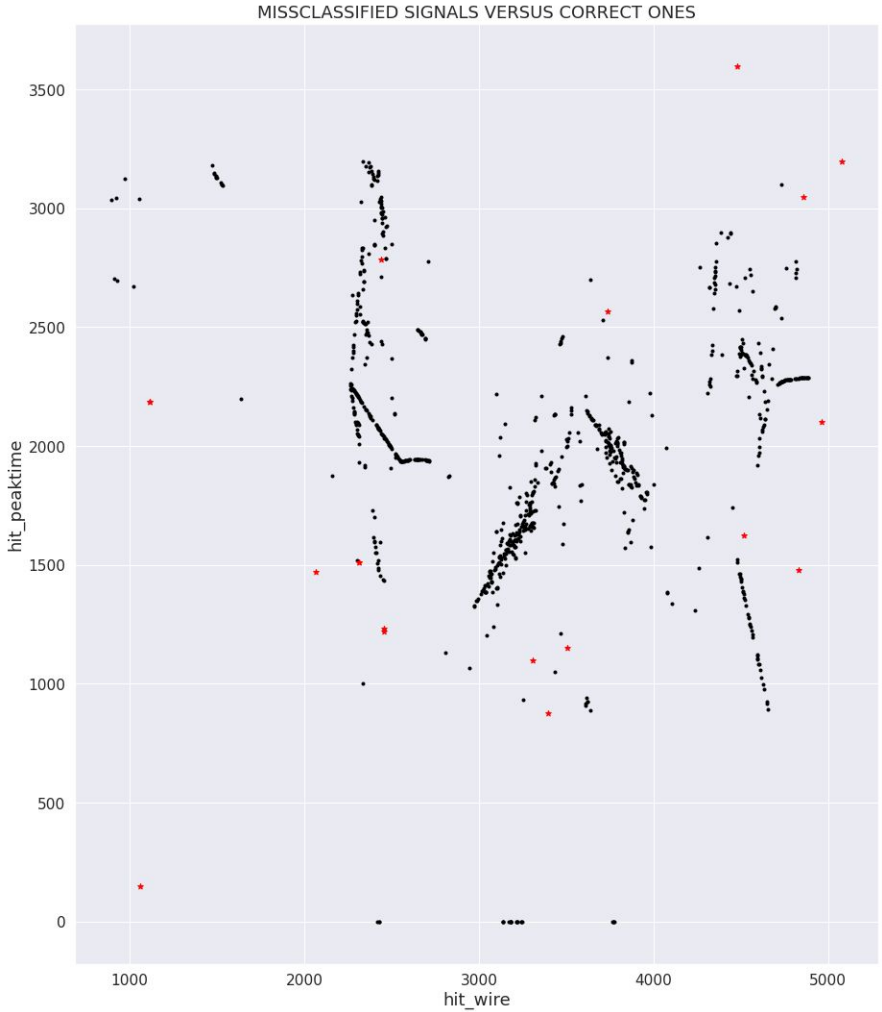


Scatter plots show classified correctly (black) and misclassified (red) nues by the algorithm.

hit_tpc = 0 & hit_plane = 2



hit_tpc = 0 & hit_plane = 2



hit_tpc = 0 & hit_plane = 2

Misclassified backgrounds that were predicted as signals; Induction

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
	71	104	0	2	1904	3599	2613.755127	0	cos
	153	1676	0	2	1804	2186	2262.582031	0	cos
./cosmicsintime_redux_10000.root	168	317	0	2	921	1169	2976.244385	0	cos
	365	292	0	2	297	2122	928.974182	0	cos
	378	742	0	2	1100	4576	1151.857788	0	cos
...	...	...	...	...	...	...	...	...	...
	51	313	0	2	613	387	2552.041748	0	cos
	76	1209	0	2	1658	879	2502.896729	0	cos
./cosmicsintime_redux_19750.root	213	362	0	2	363	2378	1340.174072	0	cos
	246	1132	0	2	1219	2323	3844.225830	0	cos
		1135	0	2	1219	2605	3063.466064	0	cos

274 rows × 7 columns

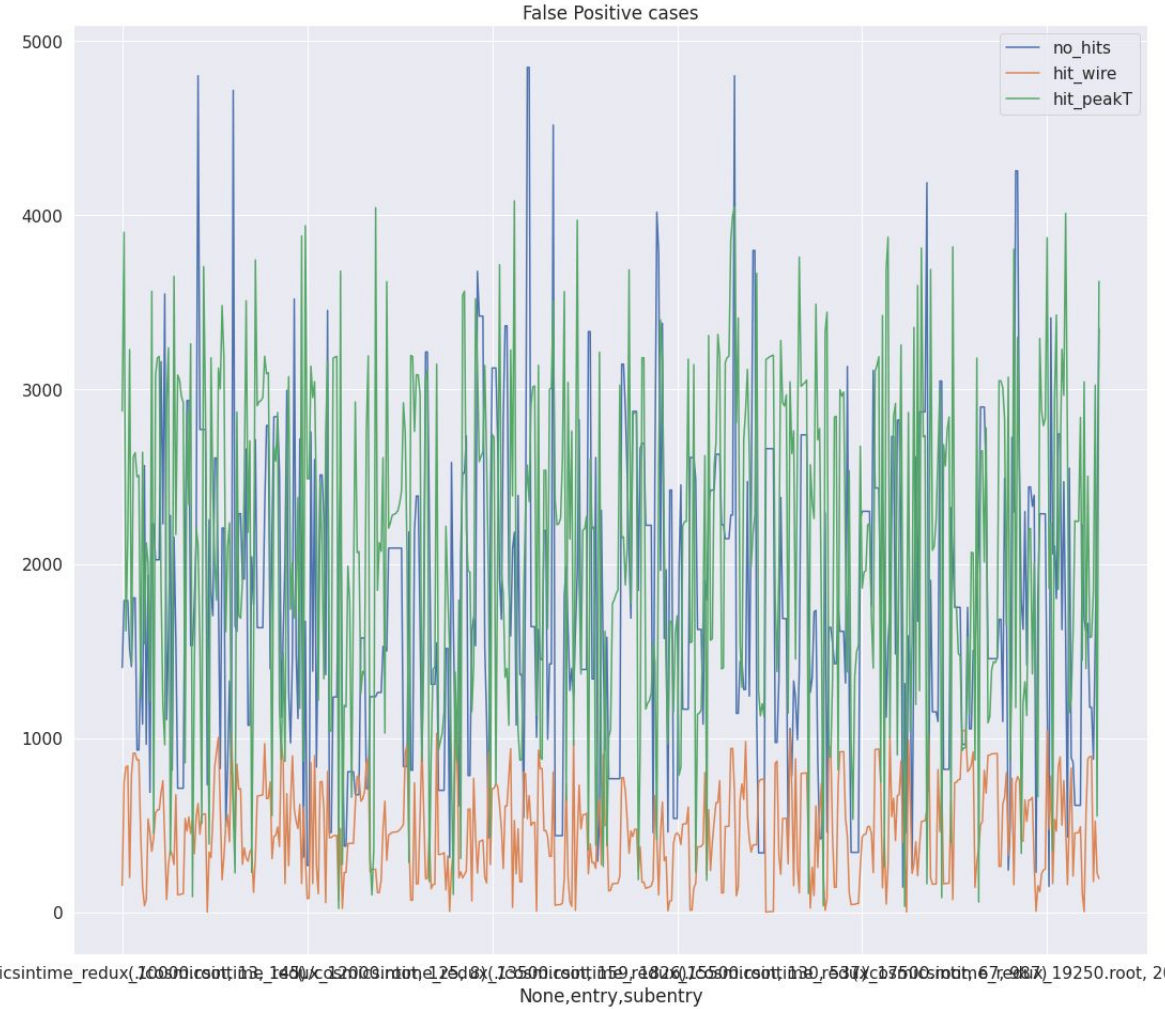
hit_tpc = 0 & hit_plane = 2

Cosmic cascades were predicted as Neutrino jets by RF

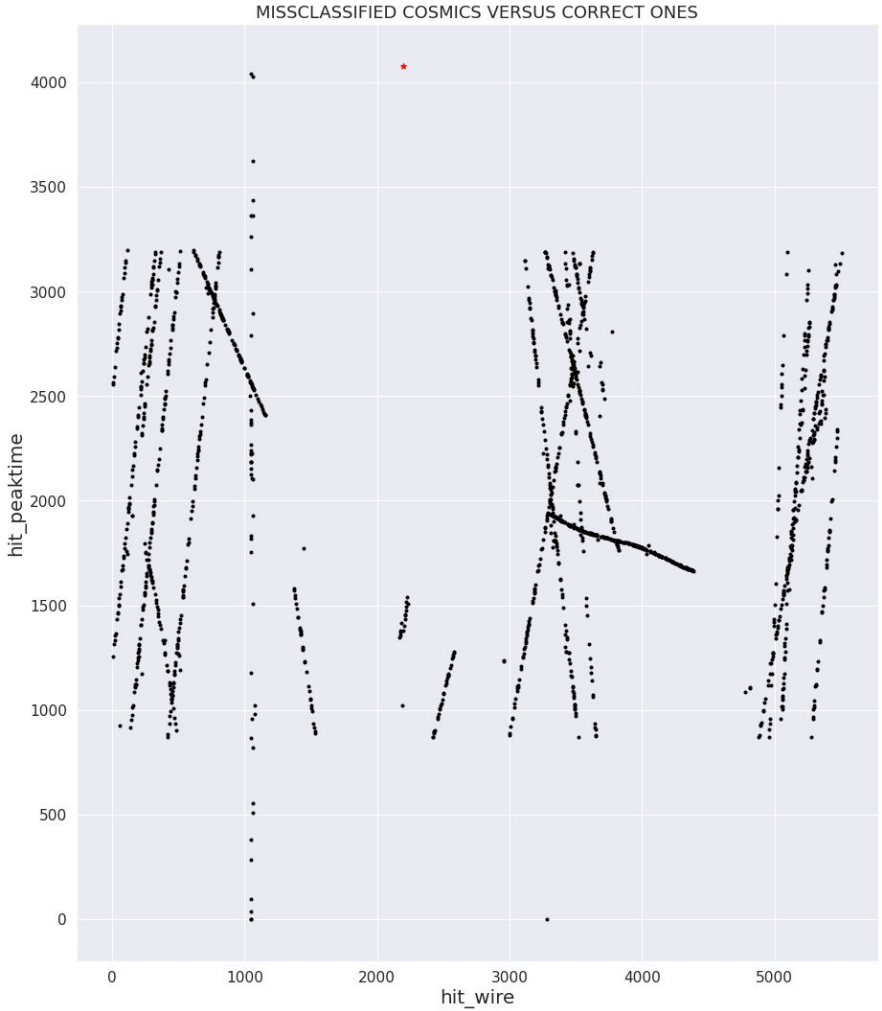
The points with same color belong to same entry

hit_tpc = 0 & hit_plane = 2

Cosmic hits predicted as Neutrino hits by RF

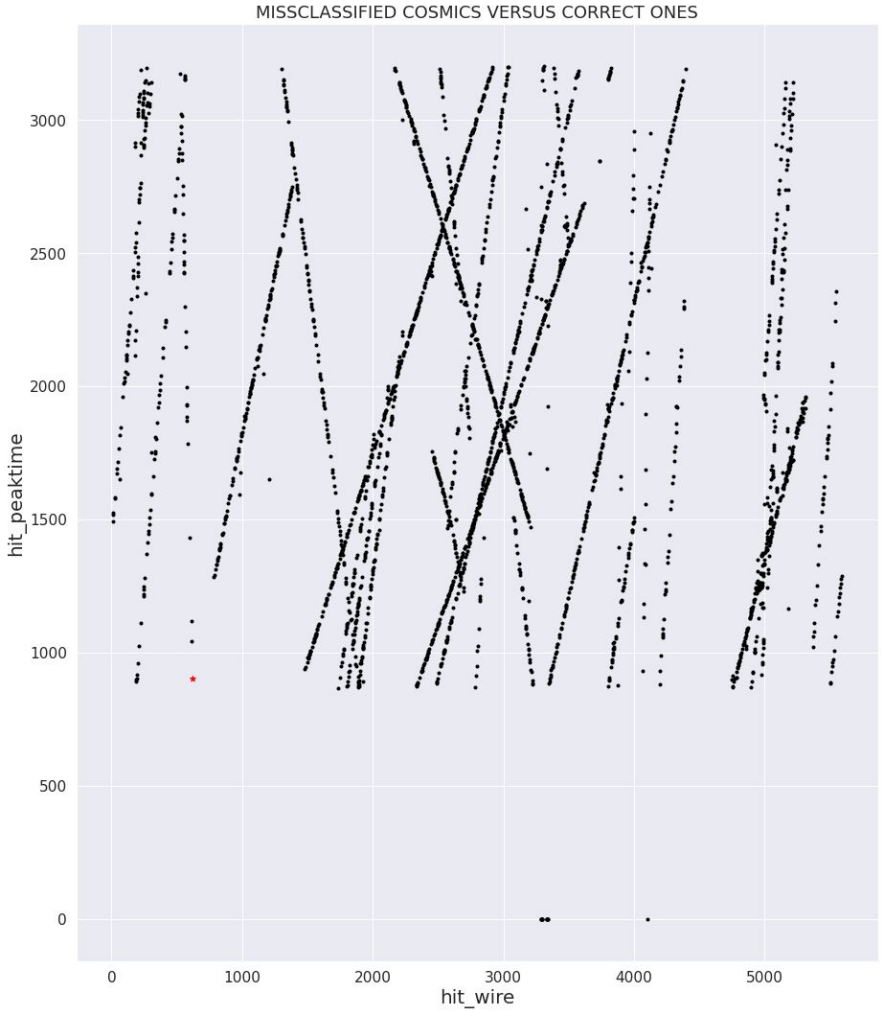


hit_tpc = 0 & hit_plane = 2

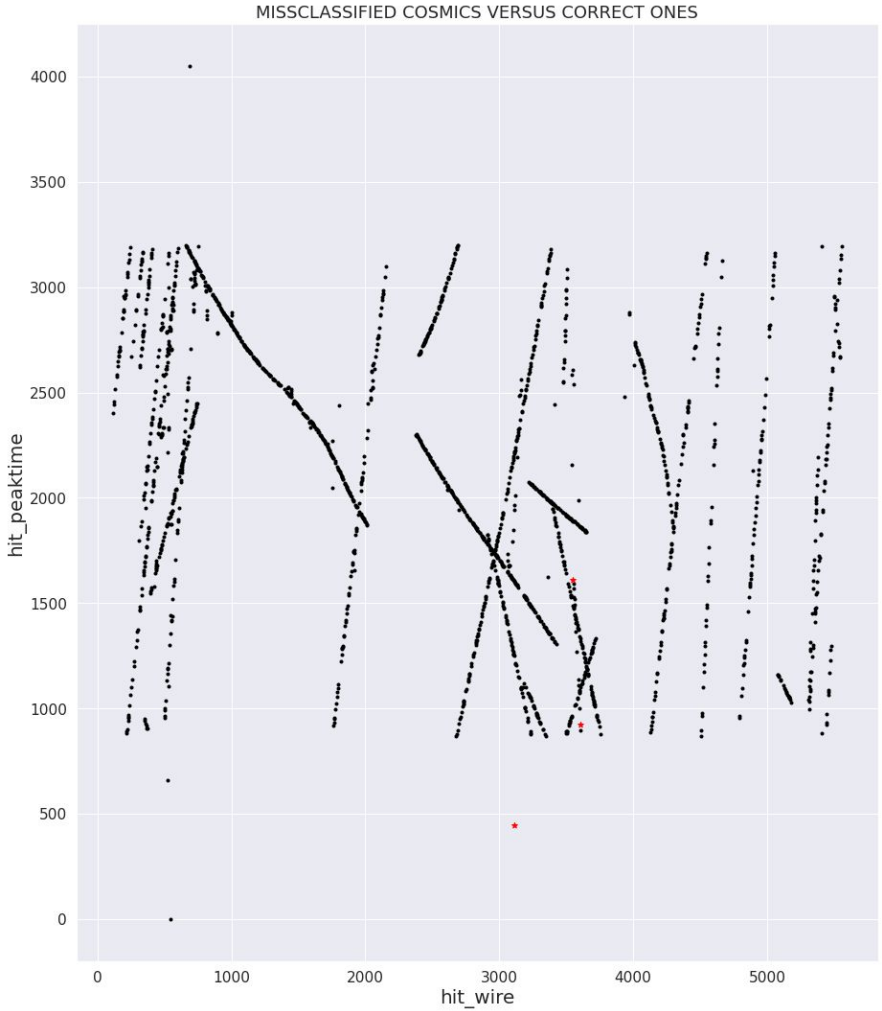


Scatter plots show classified (black) and misclassified (red) cosmics by the algorithm.

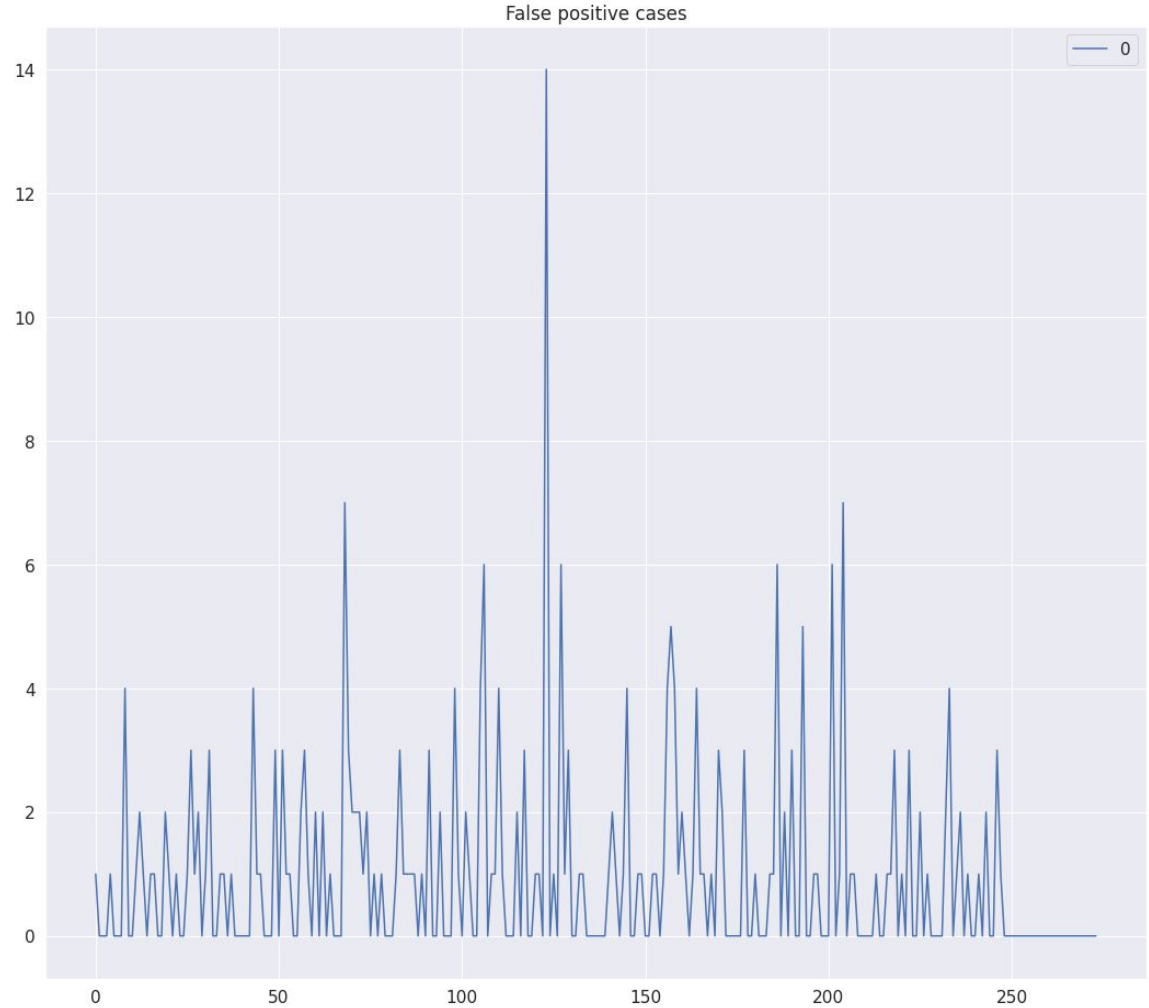
hit_tpc = 0 & hit_plane = 2



hit_tpc = 0 & hit_plane = 2



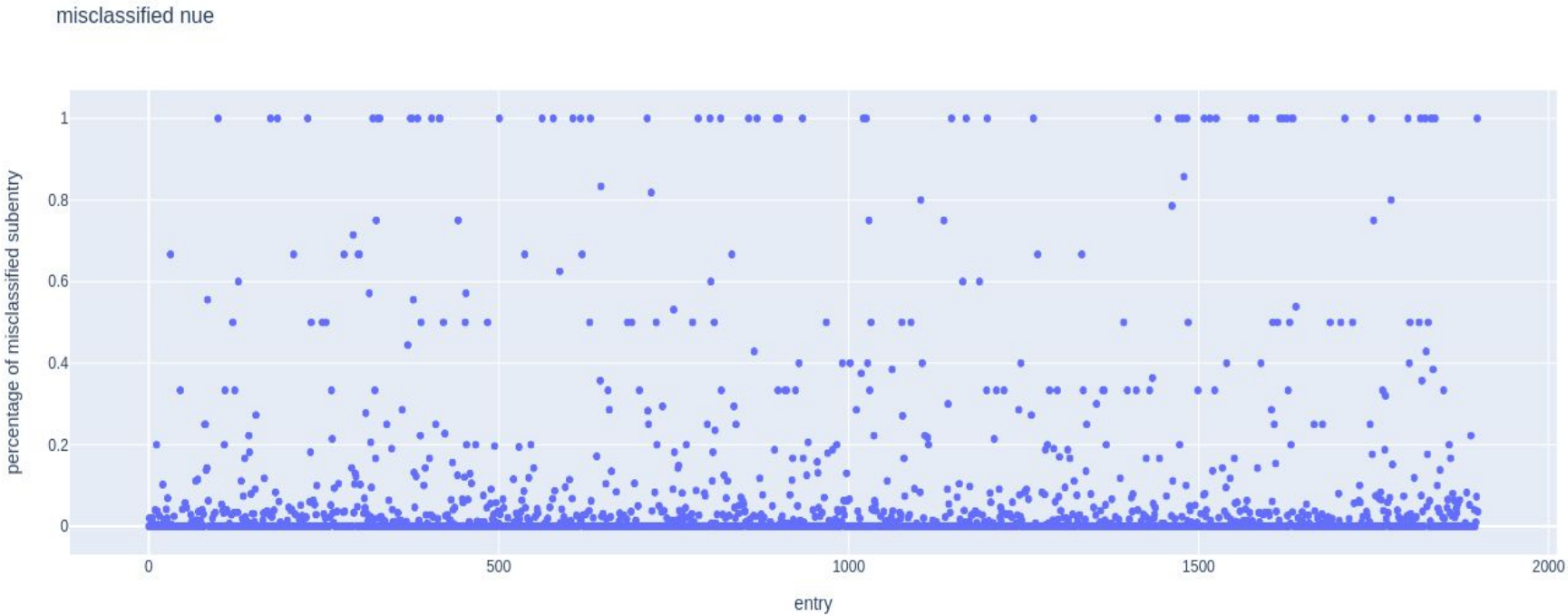
hit_tpc = 0 & hit_plane = 2



Distribution of false subentry of each entry for misclassified cosmics

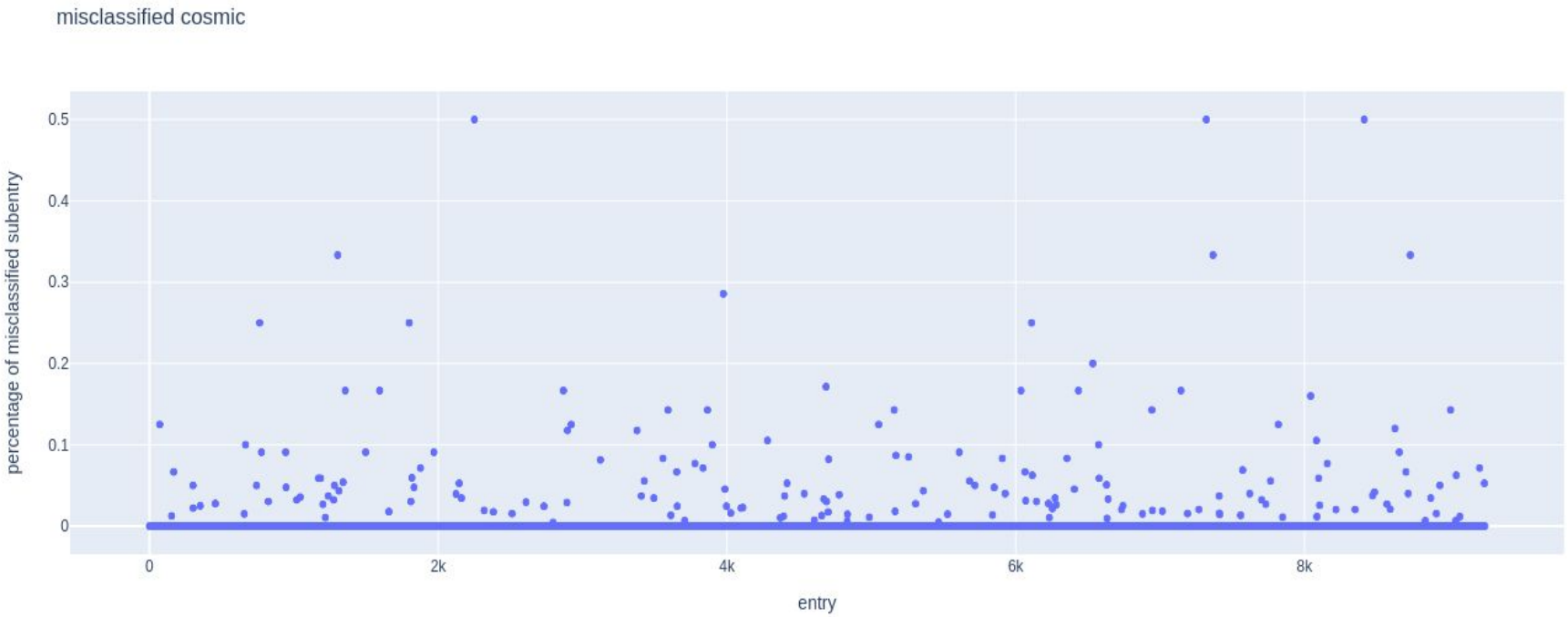
hit_tpc = 0 & hit_plane = 2

❖ Plot : Percentage of Misclassified subentry(single hits) for each entry - nue



hit_tpc = 0 & hit_plane = 2

❖ Plot : Percentage of Misclassified subentry(single hits) for each entry - cosmic



hit_tpc = 0 & hit_plane = 2

```
true negative(TN):=> 456839 || false positive(FP) :=> 274
================================================================
false negative(FN):=> 2418  || true positive(TP) :=> 76176
----------------------------------------------------------------
P => 78594 N => 457113
----------------------------------------------------------------
```

Sensitivity or true positive rate(TPR) ==> 0.9692342926940988

MISS RATE or false negative rate(FNR) ==> 0.030765707305901224

---

specificity, selectivity or true negative rate(TNR) ==> 0.9994005858507634

fall-out or false positive rate (FPR) ==> 0.0005994141492365879

---

precision or positive predictive value(PPV) ==> 0.9964159581425769

false discovery rate (FDR) ==> 0.0035840418574231148

---

negative predictive value(NPV) ==> 0.9947349740994693

false omission rate (FOR) ==> 0.005265025900530684

---

threat score(TS) or critical success index(CSI) ==> 0.9658670183090734

/\//\/\/\/\/\/\/\//\/\/\//\/\/\/\/\/\//

prevalence threshold(PT) ==> 0.024265038013227522

hit_tpc = 0 & hit_plane = 2

```
true negative(TN):=> 9421 || false positive(FP) :=> 79
=================================================================
false negative(FN):=> 424  || true positive(TP) :=> 1476
-----------------------------------------------------------------
P => 1900 N => 9500
-----------------------------------------------------------------
Sensitivity or true positive rate(TPR) ==> 0.7768421052631579
MISS RATE or false negative rate(FNR) ==> 0.2231578947368421
_________________________________________________________________
specificity, selectivity or true negative rate(TNR) ==> 0.9916842105263158
fall-out or false positive rate (FPR) ==> 0.008315789473684165
_________________________________________________________________
precision or positive predictive value(PPV) ==> 0.9491961414790997
false discovery rate (FDR) ==> 0.05080385852090032
_________________________________________________________________
negative predictive value(NPV) ==> 0.9569324530218385
false omission rate (FOR) ==> 0.04306754697816151
_________________________________________________________________
threat score(TS) or critical success index(CSI) ==> 0.7458312278928751
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\//\/\/\/\/\//\/\/\//\/\/\/\
prevalence threshold(PT) ==> 0.09376215086377451
=================================================================
=================================================================
Accuracy of the model at entry level(ACC) ==> 0.9558771929824561
=================================================================
=================================================================
```

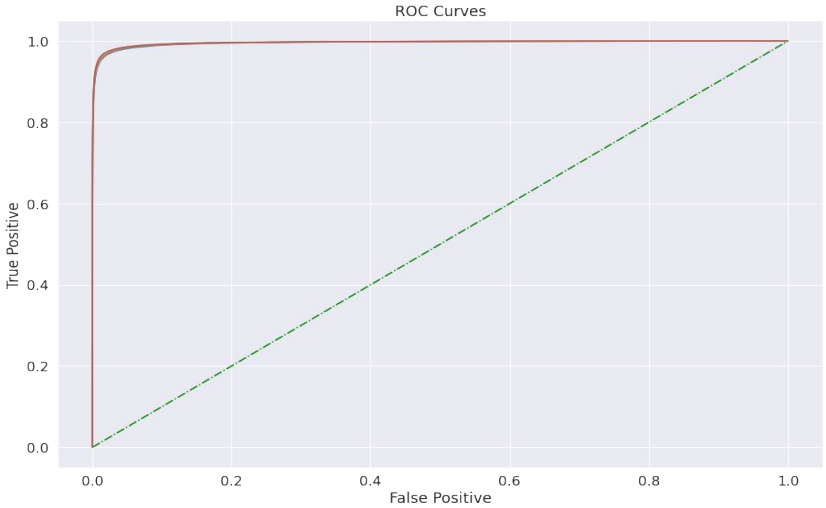
➔ Performance of the model in entry level for collection hits

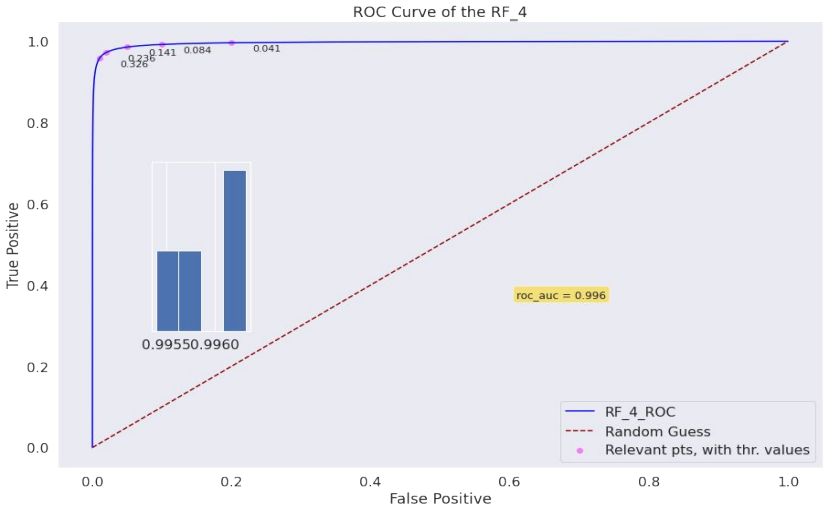
hit_tpc = 0 & hit_plane = 2

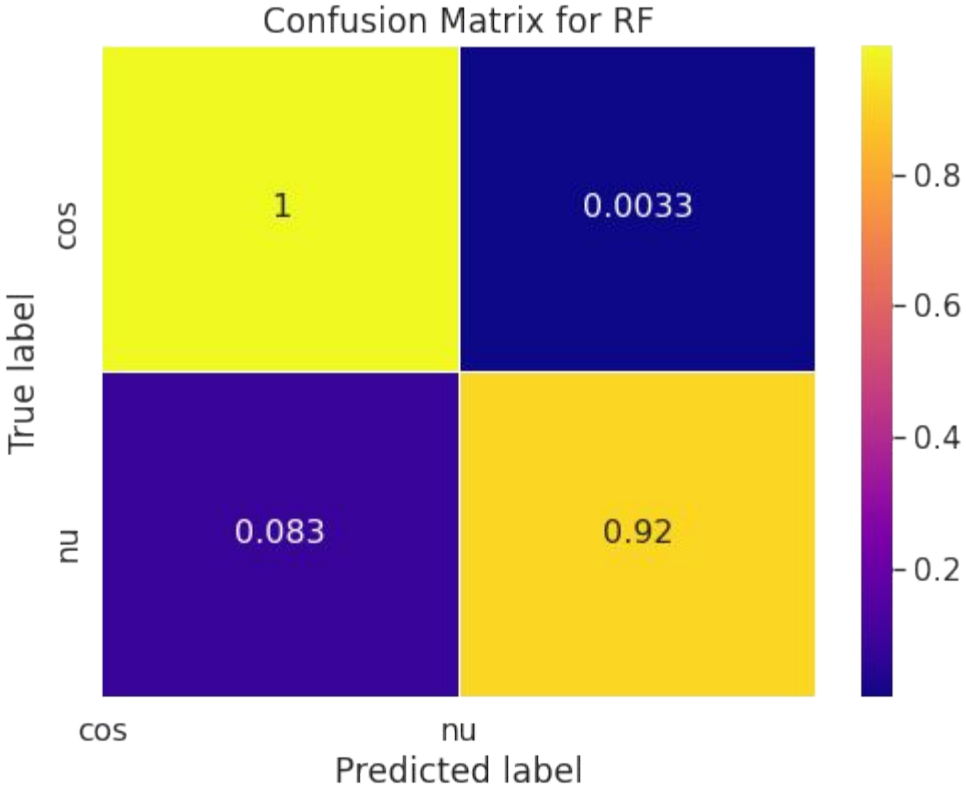
hit_tpc = 1 & hit_plane = 1

values	0.0100	0.0200	0.0500	0.1000	0.2000
False Positive	0.0100	0.0200	0.0500	0.1000	0.2000
True Positive	0.9577	0.9726	0.9859	0.9920	0.9963
threshold	0.3261	0.2365	0.1408	0.0843	0.0406

```
root mean square error for Random Forest 12.190119072881467
accuracy of random forest 0.9851400996988972
```







hit_tpc = 1 & hit_plane = 1

Confusion Matrix for
Collection zone data

```
[[475514   1583]
 [  6713  74471]]
```

Confusion Matrix for
Collection zone data

hit_tpc = 1 & hit_plane = 1

```
[[475514   1583]
 [  6713  74471]]
```

```
true negative(TN):=> 475514 || false positive(FP) :=> 1583
=================================================================
false negative(FN):=> 6713  || true positive(TP) :=> 74471
-----------------------------------------------------------------
P => 81184 N => 477097
-----------------------------------------------------------------
Sensitivity or true positive rate(TPR) ==> 0.9173112928655893
MISS RATE or false negative rate(FNR) ==> 0.08268870713441068
_________________________________________________________________
specificity, selectivity or true negative rate(TNR) ==> 0.9966820164452931
fall-out or false positive rate (FPR) ==> 0.003317983554706916
_________________________________________________________________
precision or positive predictive value(PPV) ==> 0.9791858416388356
false discovery rate (FDR) ==> 0.020814158361164403
_________________________________________________________________
negative predictive value(NPV) ==> 0.9860791701833369
false omission rate (FOR) ==> 0.013920829816663094
_________________________________________________________________
threat score(TS) or critical success index(CSI) ==> 0.89976681527662
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\//\/\/\/\/\/\//\/\/\//\/\/\/\/\/\//
prevalence threshold(PT) ==> 0.056730244363850976
```

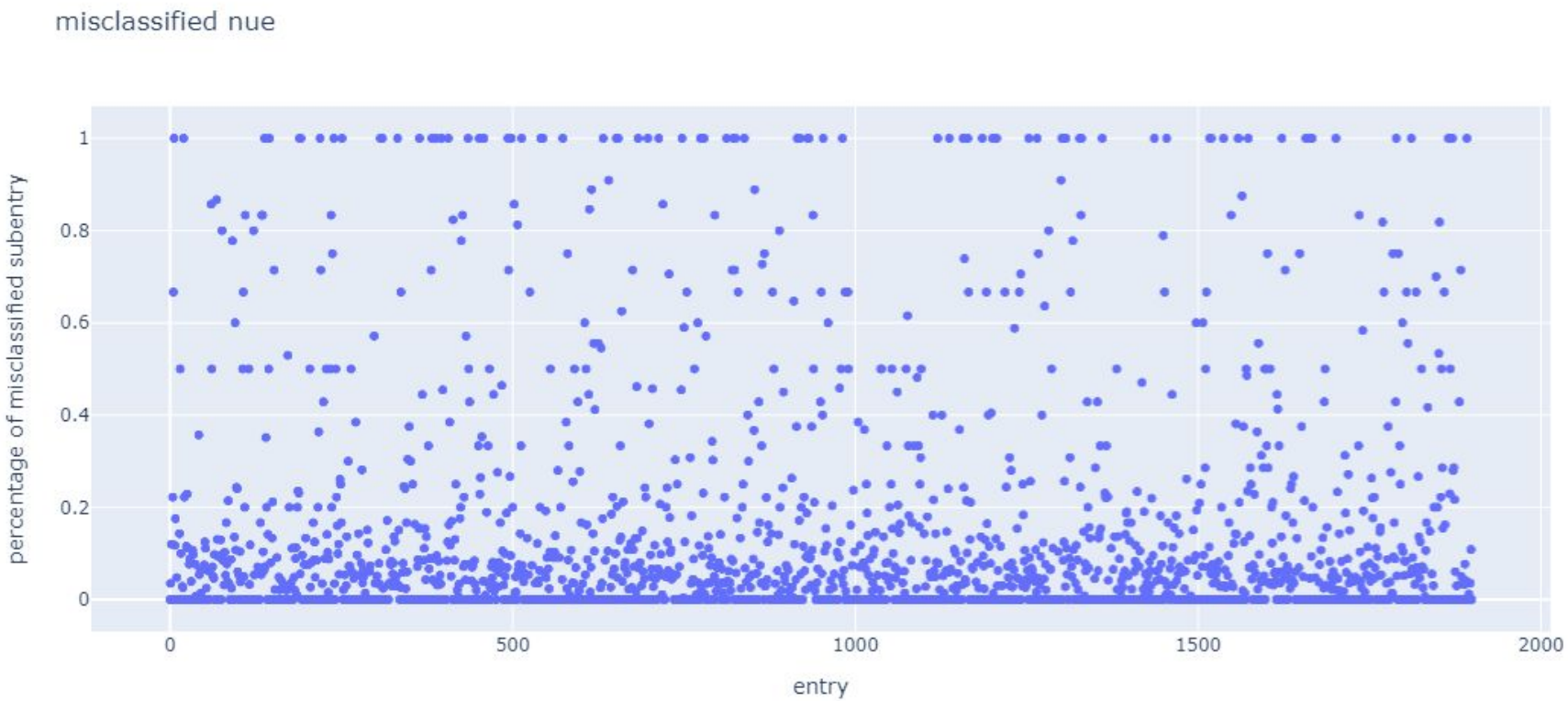
Misclassified nues that were predicted as cosmics;

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
	1	820	1	1	1369	874	2210.0	1	ev
		1074	1	1	1369	1118	1815.0	1	ev
./nue_0.root		193	1	1	954	4496	1105.0	1	ev
	2	380	1	1	954	4719	980.0	1	ev
		490	1	1	954	4823	936.0	1	ev
...	...	...	...	...	...	...	...	...	...
		1313	1	1	2229	5323	2320.0	1	ev
		1319	1	1	2229	5356	295.0	1	ev
./nue_900.root	98	1320	1	1	2229	5452	2935.0	1	ev
		1327	1	1	2229	5460	2955.0	1	ev
		1334	1	1	2229	5467	2834.0	1	ev

7116 rows × 7 columns

hit_tpc = 1 & hit_plane = 1

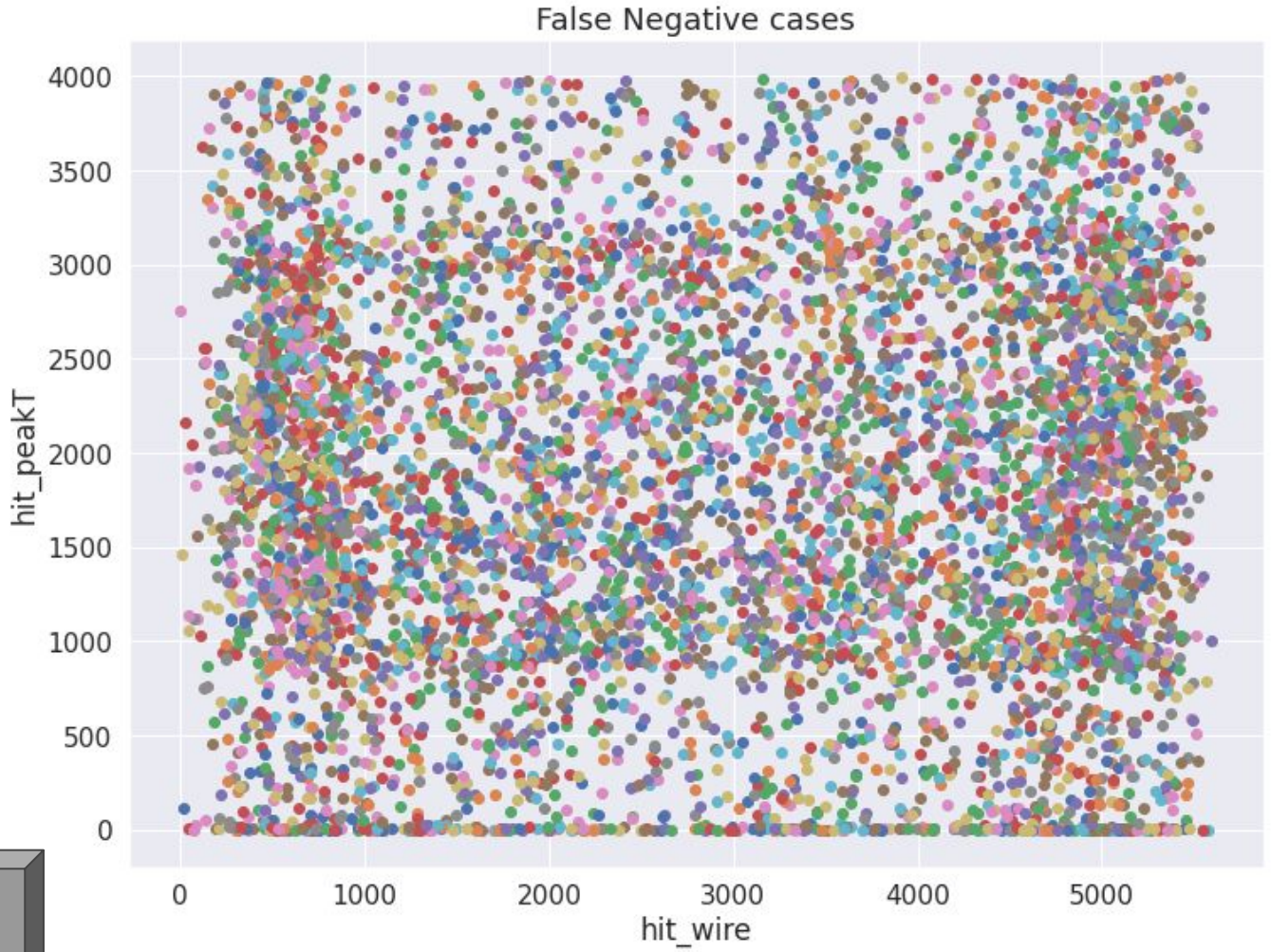
❖ Plot : Percentage of Misclassified subentry(single hits) for each entry - nue



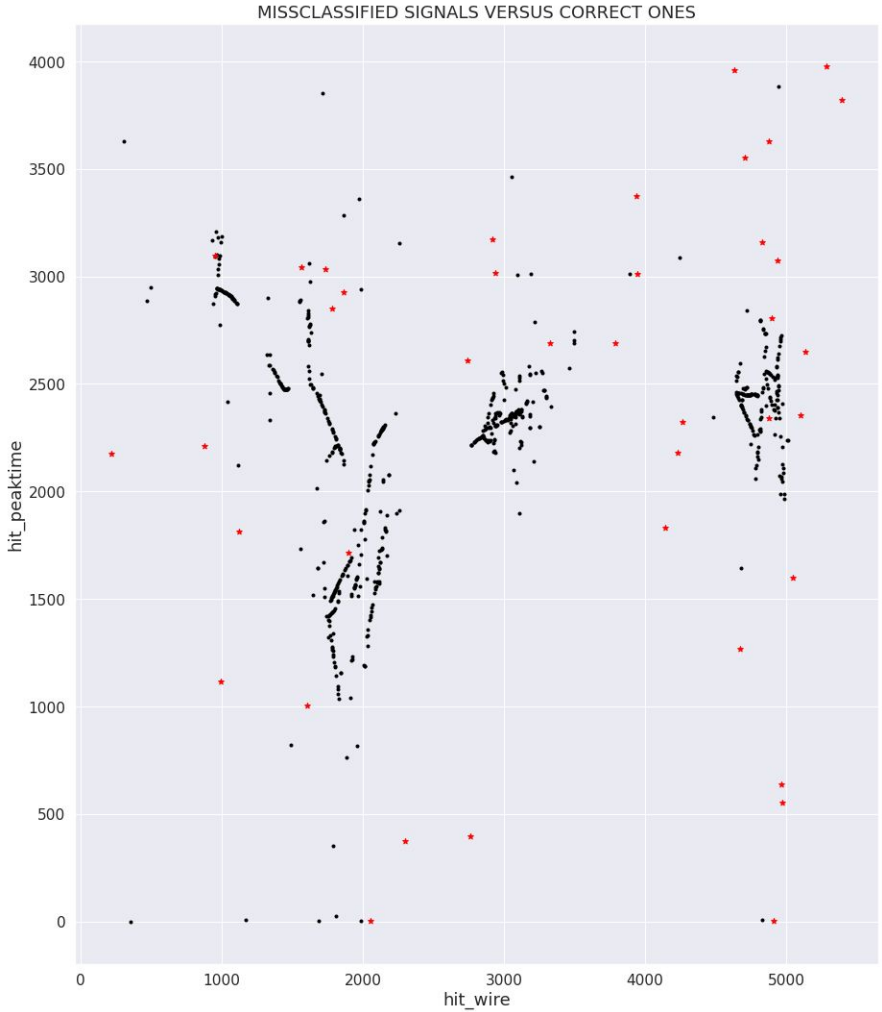
hit_tpc = 1 & hit_plane = 1

Neutrino tracks were predicted as Cosmic by RF

The points with same color belong to same entry

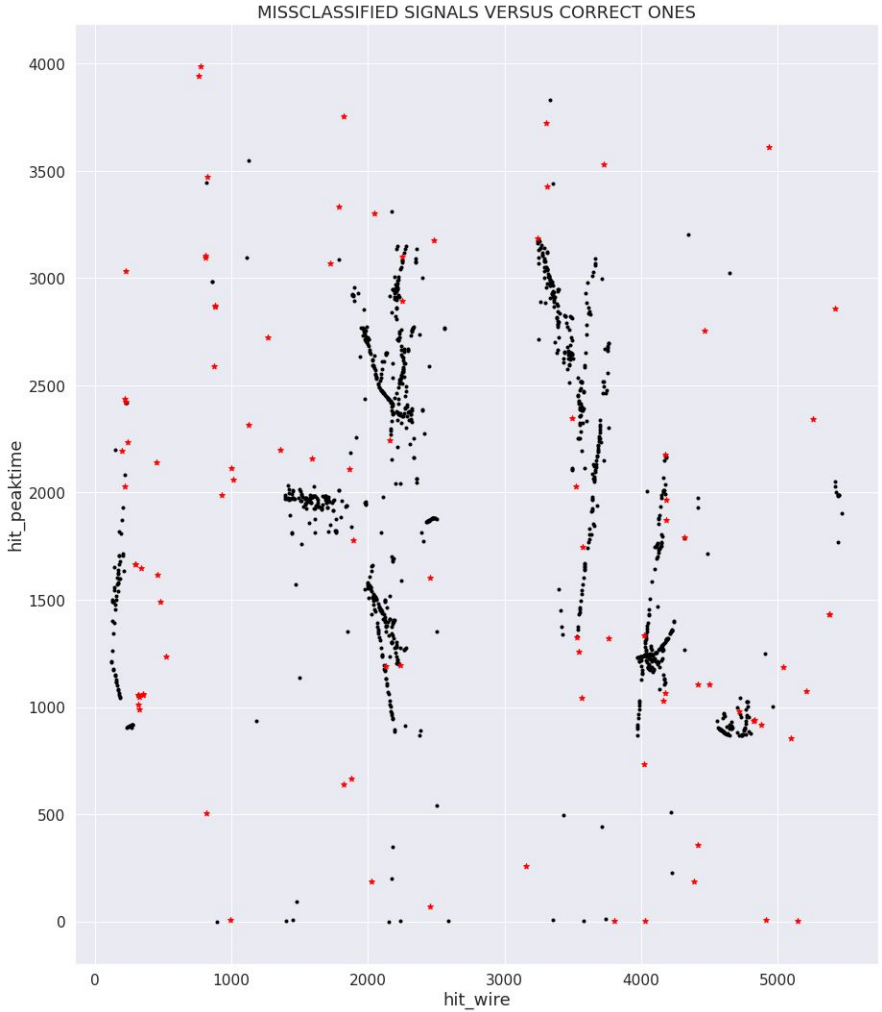


hit_tpc = 1 & hit_plane = 1

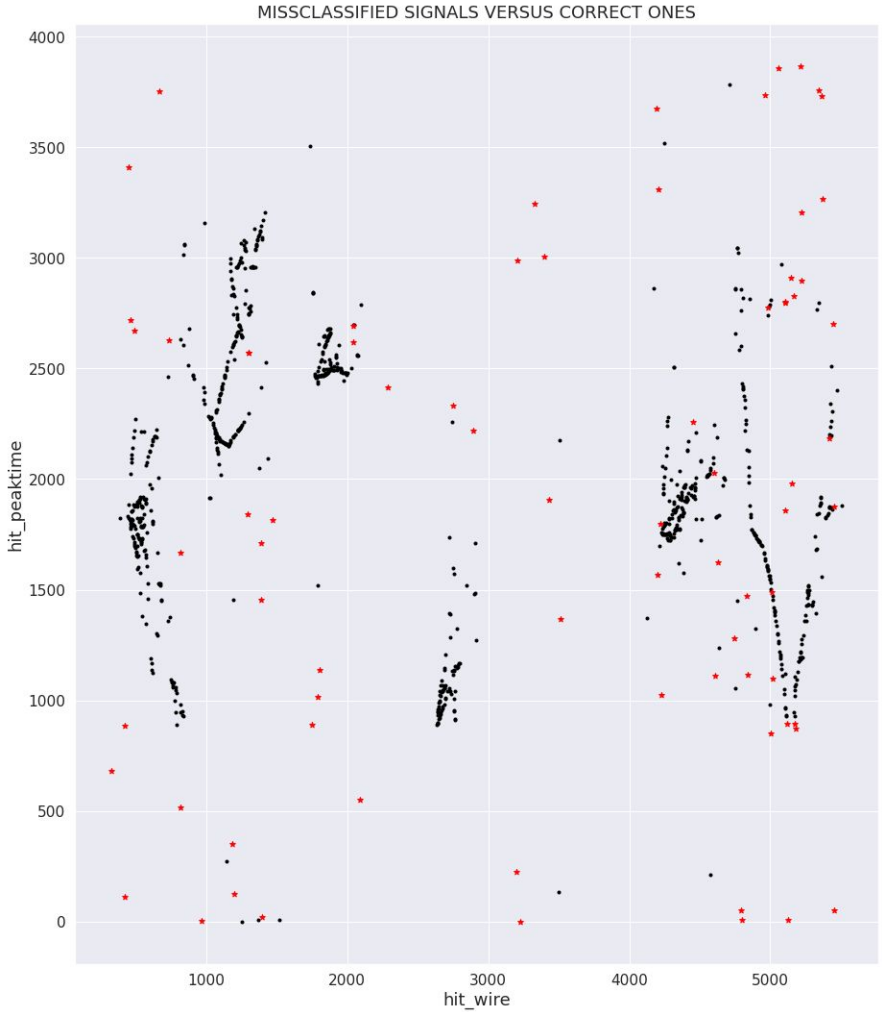


Scatter plots indicate classified (black) and misclassified (red) nues by the algorithm.

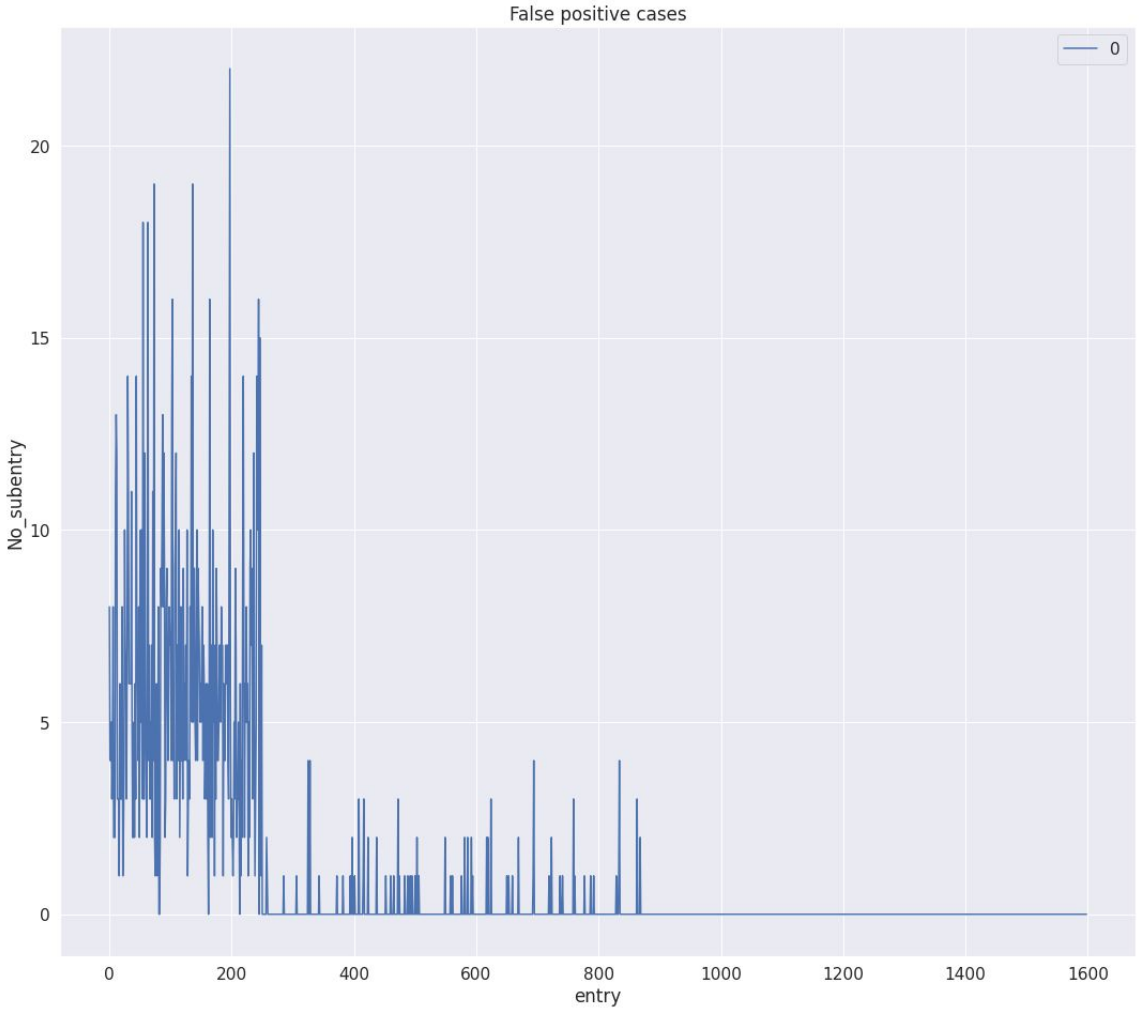
hit_tpc = 1 & hit_plane = 1



hit_tpc = 1 & hit_plane = 1



hit_tpc = 1 & hit_plane = 1



Distribution of false subentry of each entry for misclassified nues

hit_tpc = 1 & hit_plane = 1

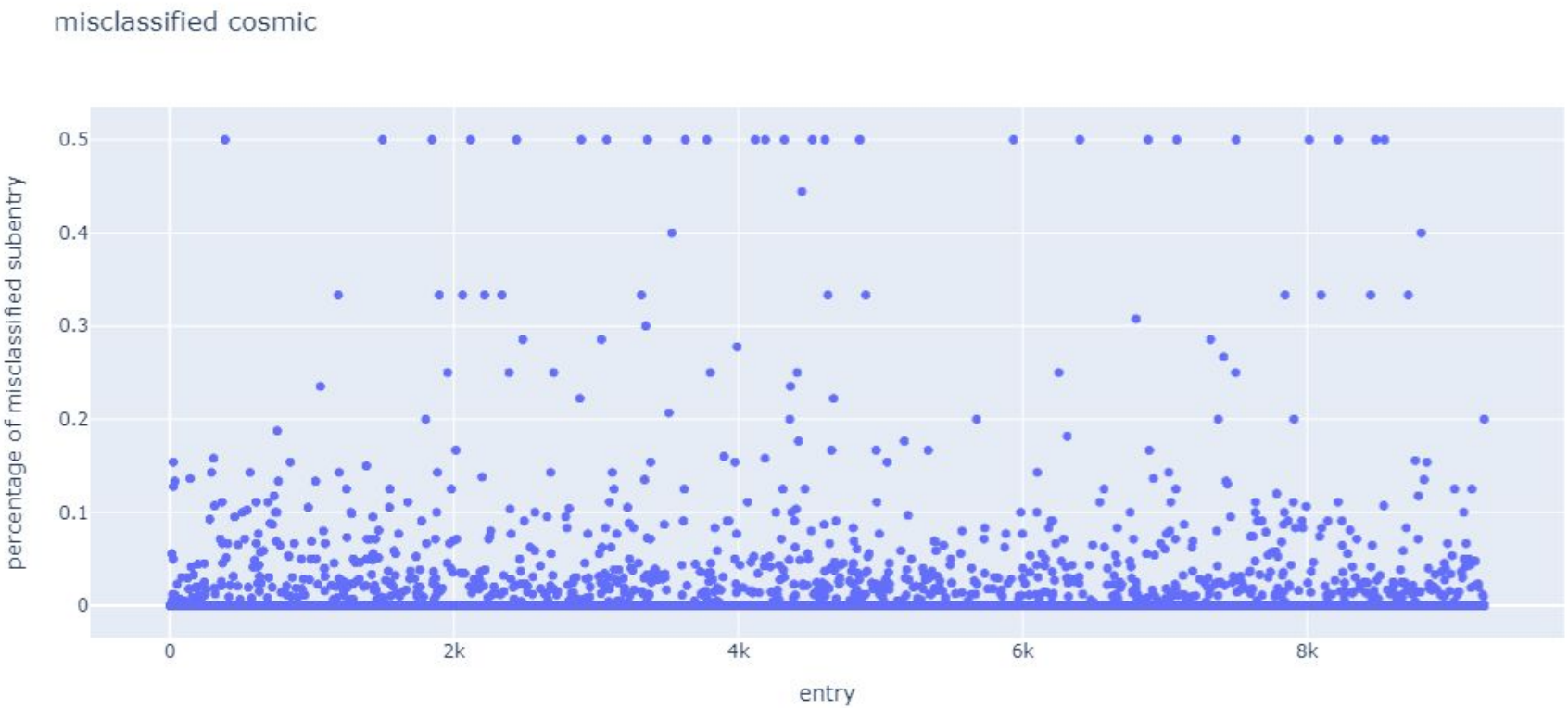
Misclassified backgrounds that were predicted as signals;

	entry	subentry	hit_tpc	hit_plane	no_hits	hit_wire	hit_peakT	label	tagg
	14	237	1	1	305	4792	138.0	0	cos
	17	1302	1	1	1721	2735	299.0	0	cos
./cosmicsintime_redux_10000.root	19	1093	1	1	1306	632	2049.0	0	cos
	24	108	1	1	179	5098	2949.0	0	cos
	25	795	1	1	1389	4130	1269.0	0	cos
...	...	...	...	...	...	...	...	...	...
	241	848	1	1	1656	87	1660.0	0	cos
	246	1180	1	1	1219	2305	2875.0	0	cos
./cosmicsintime_redux_19750.root		1191	1	1	1219	2519	3166.0	0	cos
		1192	1	1	1219	2520	3164.0	0	cos
		1194	1	1	1219	2560	3685.0	0	cos

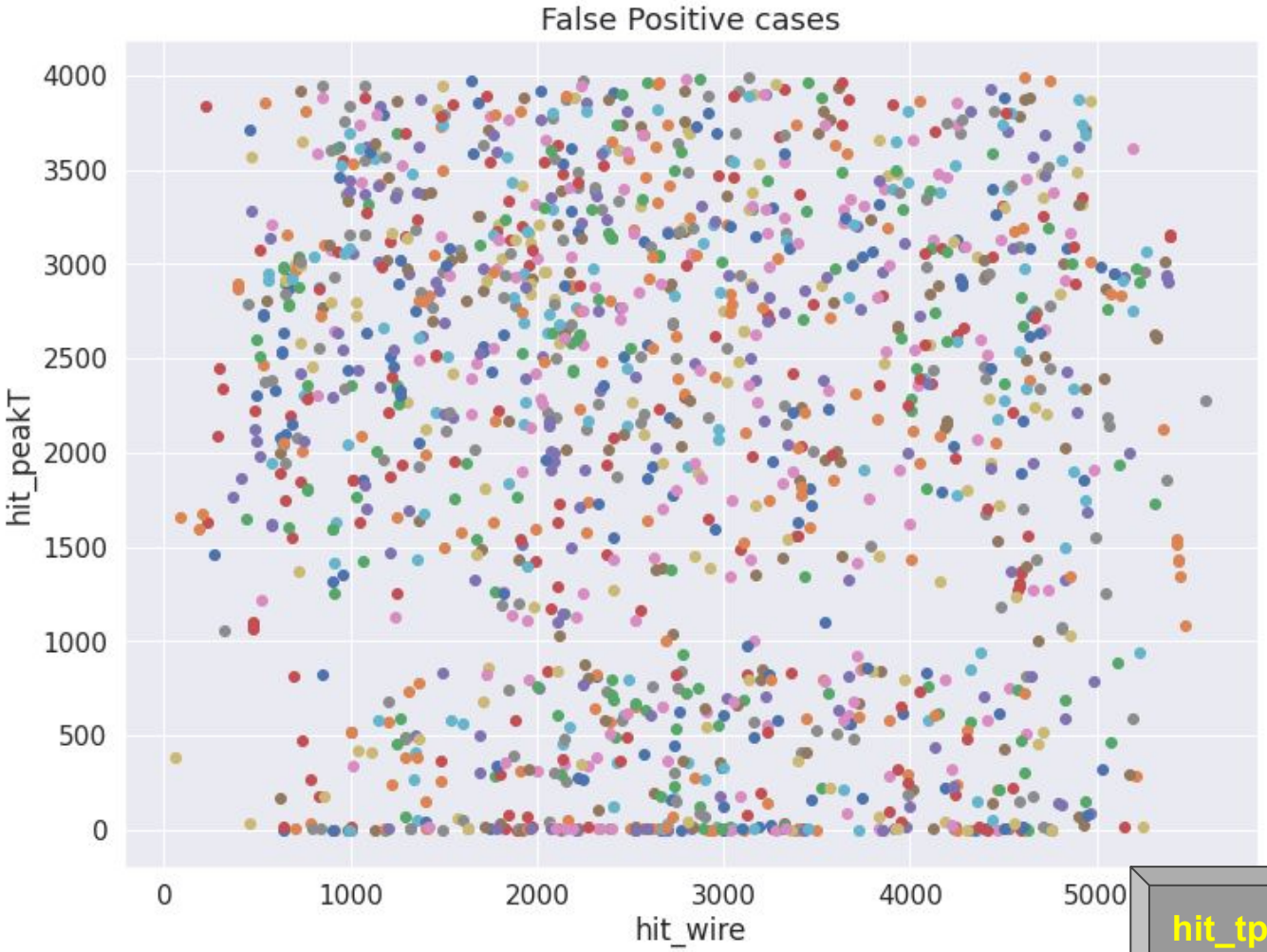
1654 rows × 7 columns

hit_tpc = 1 & hit_plane = 1

❖ Plot : Percentage of Misclassified subentry(single hits) for each entry - cosmic



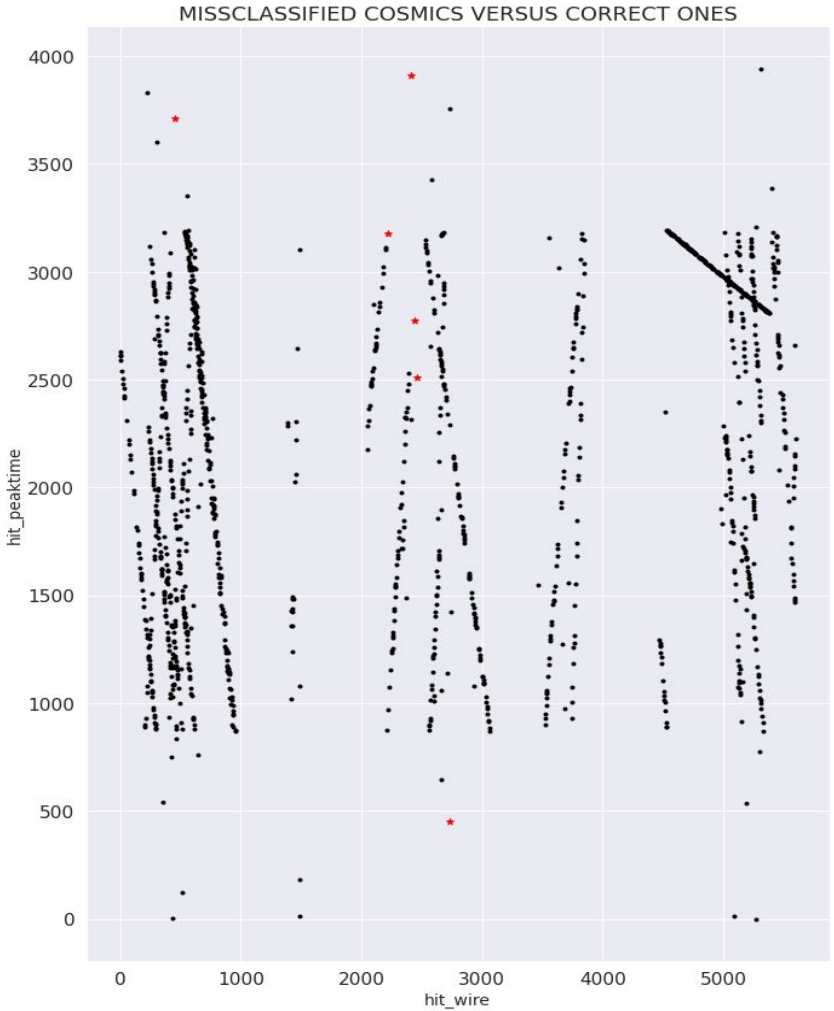
hit_tpc = 1 & hit_plane = 1



Misclassified cosmics

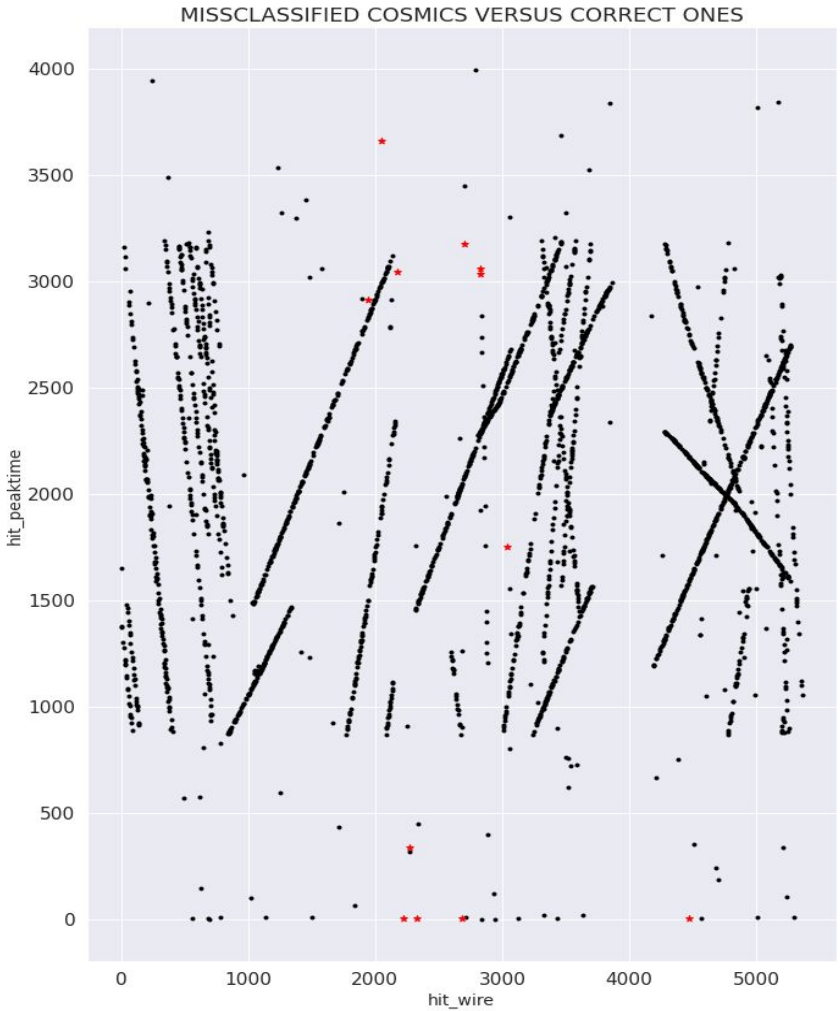
The points with same color belong to same entry

hit_tpc = 1 & hit_plane = 1

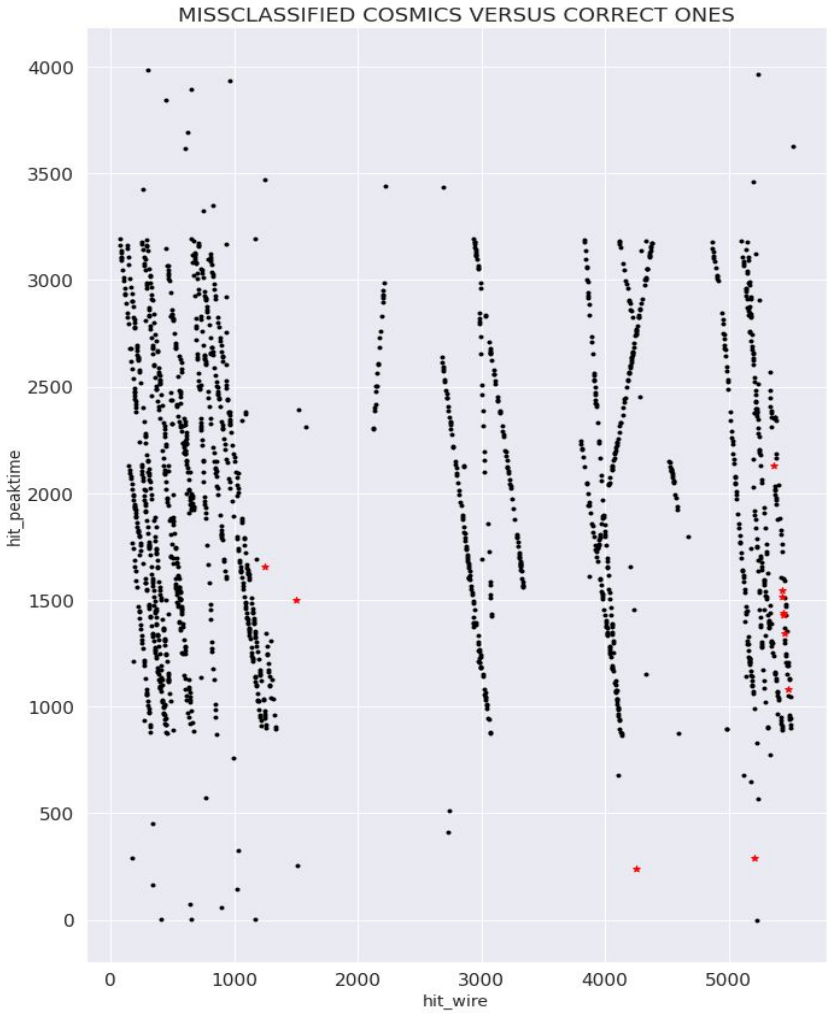


Scatter plots show classified (black) and misclassified (red) cosmics by the algorithm.

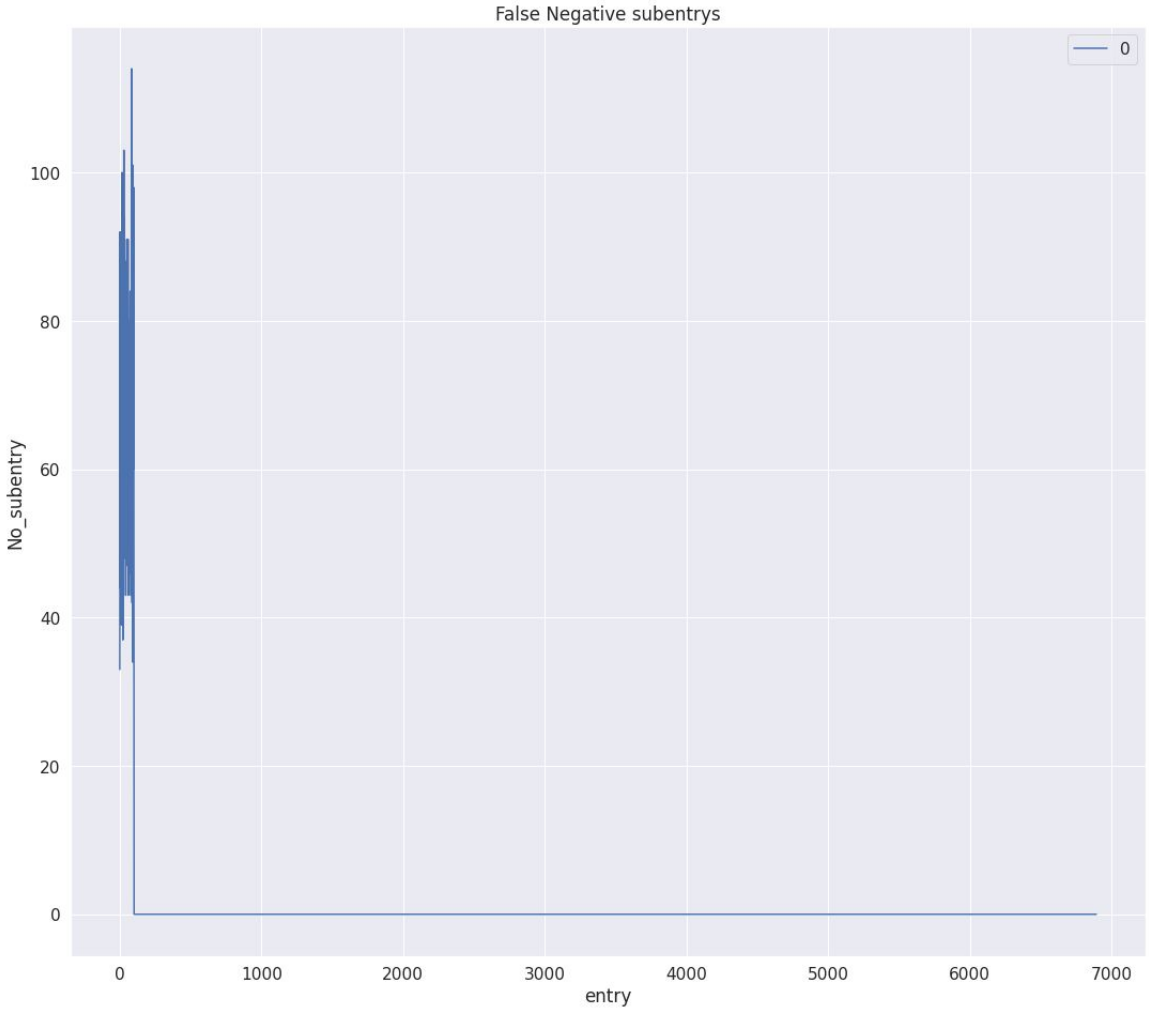
hit_tpc = 1 & hit_plane = 1



hit_tpc = 1 & hit_plane = 1



hit_tpc = 1 & hit_plane = 1



Distribution of false subentry of each entry for misclassified cosmics

hit_tpc = 1 & hit_plane = 1

Confusion Matrix for
Collection zone data

hit_tpc = 1 & hit_plane = 1

```
[[475514   1583]
 [  6713  74471]]
```

```
true negative(TN):=> 475514 || false positive(FP) :=> 1583
=================================================================
false negative(FN):=> 6713  || true positive(TP) :=> 74471
-----------------------------------------------------------------
P => 81184 N => 477097
-----------------------------------------------------------------
Sensitivity or true positive rate(TPR) ==> 0.9173112928655893
MISS RATE or false negative rate(FNR) ==> 0.08268870713441068
________________________________________________________________
specificity, selectivity or true negative rate(TNR) ==> 0.9966820164452931
fall-out or false positive rate (FPR) ==> 0.003317983554706916
________________________________________________________________
precision or positive predictive value(PPV) ==> 0.9791858416388356
false discovery rate (FDR) ==> 0.020814158361164403
________________________________________________________________
negative predictive value(NPV) ==> 0.9860791701833369
false omission rate (FOR) ==> 0.013920829816663094
________________________________________________________________
threat score(TS) or critical success index(CSI) ==> 0.89976681527662
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\//\/\/\/\/\/\/\//\/\/\//\/\/\/\/\/\//
prevalence threshold(PT) ==> 0.056730244363850976
```

```
true negative(TN):=> 8924 || false positive(FP) :=> 325
================================================================
false negative(FN):=> 952  || true positive(TP) :=> 948
----------------------------------------------------------------
P => 1900 N => 9249
----------------------------------------------------------------
Sensitivity or true positive rate(TPR) ==> 0.49894736842105264
MISS RATE or false negative rate(FNR) ==> 0.5010526315789474
________________________________________________________________
specificity, selectivity or true negative rate(TNR) ==> 0.9648610660611958
fall-out or false positive rate (FPR) ==> 0.035138933938804184
________________________________________________________________
precision or positive predictive value(PPV) ==> 0.7446975648075412
false discovery rate (FDR) ==> 0.2553024351924587 7
________________________________________________________________
negative predictive value(NPV) ==> 0.9036046982584042
false omission rate (FOR) ==> 0.09639530174159583
________________________________________________________________
threat score(TS) or critical success index(CSI) ==> 0.4260674157303371
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
prevalence threshold(PT) ==> 0.2097230777090594 3
================================================================
================================================================
Accuracy of the model - entry level(ACC) ==> 0.8854605794241636
================================================================
================================================================
```

➔ Performance of the model in entry level for collection hits

hit_tpc = 1 & hit_plane = 1

Outcomes

Based on results, the best RF model recorded the better efficiency to classify nue and cosmic for subentry level. The reason behind that refers to this subject that the samples or training data are single hit. Despite of that, the model turned over data from Collection (hit_plane = 2, hit_tpc = 0) granted the best performance, while the algorithm for induction 2 can predict the lower amount of single hits and also event. This might refer to this fact that Collection of ICARUS T600 has the highest resolution.

In all of cases, TNR is significantly more than TPR which tells us that the algorithm can recognize cosmic rays. It is also predictable as the size of cosmic 10 times higher than nue.

Further down the road

>> data with overlapped cosmic and neutrino hits
>> application to real data with lower signal-to-noise ratio
>> using real data with the current RF model and tuning the parameters of the model to increase the accuracy

THANK YOU