LZ pulse classification

TAAD - Final Project 04.02.2022 by Helena Lessa and Patricia Pesch

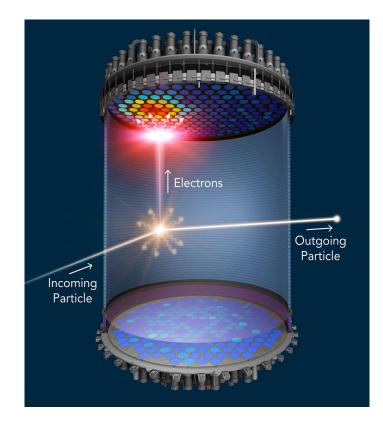
Source: Machine Learning tools for pulse classification in LZ, P. Brás, Ciência dos Dados em Física 2021

The LUX-ZEPLIN (LZ) detector

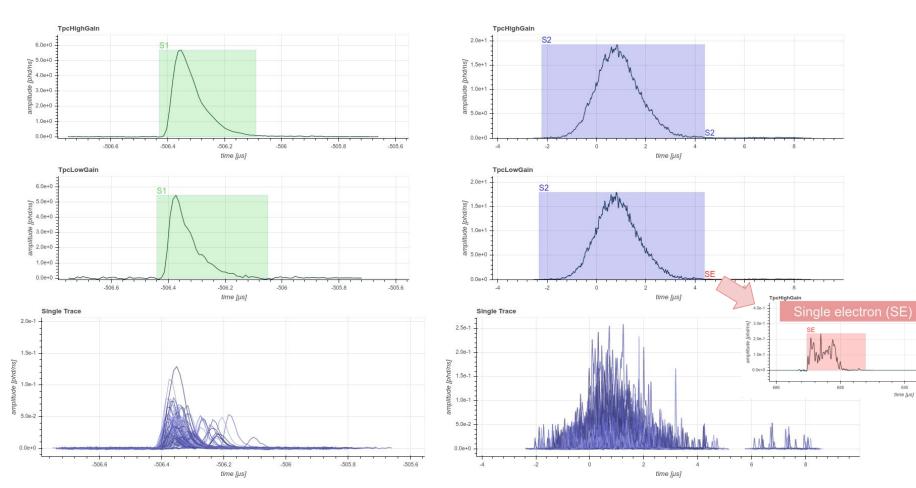
The LZ is a dark matter direct detection experiment expected to start running in 2022. It will use a two-phase xenon Time Projection Chamber (TPC) to detect Weakly Interacting Massive Particles (WIMPs).

Operating principle:

- 1. Energy deposits produces a prompt scintillation light (S1) and ionizes electrons.
- 2. Some of these electrons recombine with xenon ions, and the remaining ones drift in an electric field.
- 3. The electrons are extracted by another field into the gas region, creating **electroluminescence light (S2).** When only one electron is extracted, the signal is called **single electron (SE)**.



Different pulses



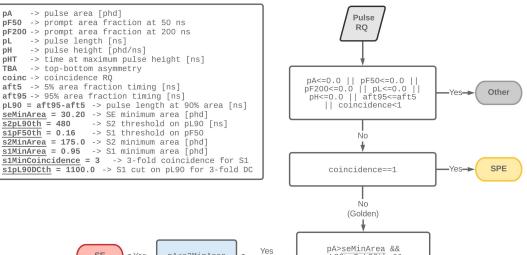
3

Current pulse classifier

HADES (Heuristic Algorithm for Discrimination of Event Substructures)

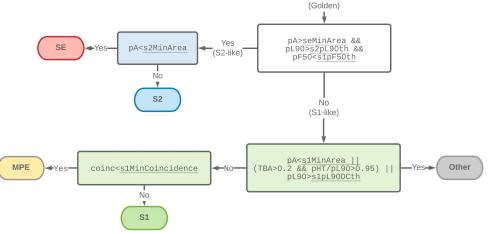
- Heuristic decision tree
- Uses only 10 features
- Trained by hand

Estimated classification accuracy: 98.6%



Boosted Decision Trees (BDTs)

Ensemble of weak learners (trees) such that each new generation of trees focuses on the misclassification of the previous generations by assigning different weights to the samples.

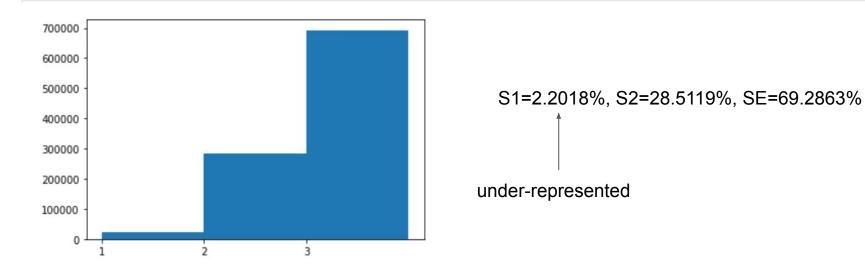


Check class representativity

Check class representativity '''

1.1.1

```
print(labels)
hist = plt.hist(labels,[1,2,3,4])
plt.xticks((1,2,3))
num_pulses_S1 = hist[0][0]
num_pulses_S2 = hist[0][1]
num_pulses_SE = hist[0][2]
print('S1={}%, S2={}%, SE={}%'.format(100*hist[0][0]/num_pulses, 100*hist[0][1]/num_pulses, 100*hist[0][2]/num_pulses)]
```

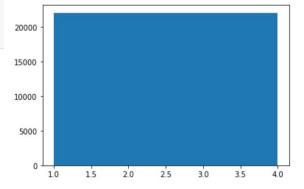


Scale down data

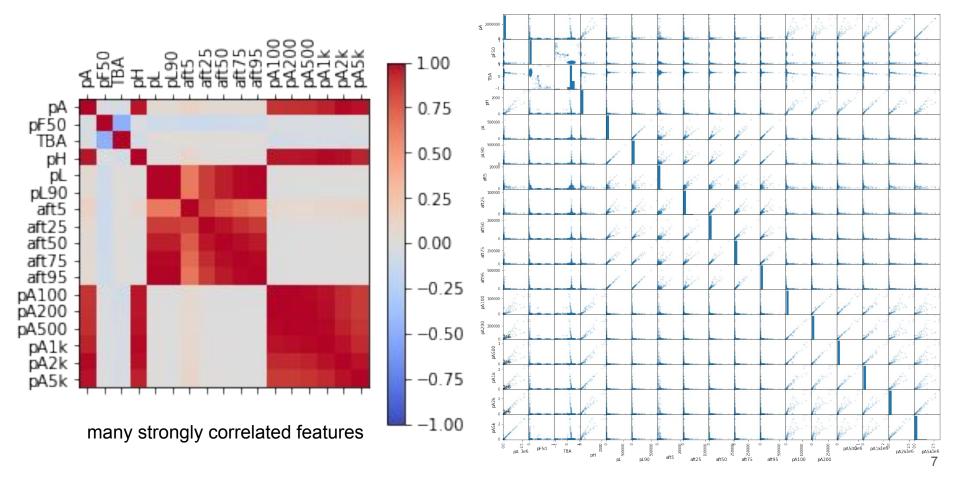
```
Scale down data because S2 and S3 are over represented
1.1.1
data_for_scaling = pd.read_csv(filename)
S1 = data_for_scaling[data_for_scaling['pulseClass'] == 1]
S2 = data_for_scaling[data_for_scaling['pulseClass'] == 2]
SE = data_for_scaling[data_for_scaling['pulseClass'] == 3]
print('Before downscaling: S1={}%, S2={}%'.format(100*len(S1)/num_pulses, 100*len(S2)/num_pulses, 100*len(SE)/1
num_pulses_min = int(min([num_pulses_S1, num_pulses_S2, num_pulses_SE]))
S1_down = S1.sample(num_pulses_min)
S2_down = S2.sample(num_pulses_min)
SE_down = SE.sample(num_pulses_min)
data_down = pd.concat([S1_down, S2_down, SE_down])
data_down = data_down.sample(frac=1).reset_index(drop=True)
num_pulses_down = len(data_down)
print('After downscaling: S1={}%, S2={}%, SE={}%'.format(100*len(S1_down)/num_pulses_down, 100*len(S2_down)/num_pulses.
print('Total number of pulses: ', num_pulses_down)
```

```
labels_down = pd.DataFrame(data_down, columns = ['pulseClass'])
hist_down = plt.hist(labels_down, [1,2,3,4])
```

Before downscaling: S1=2.2018%, S2=28.5119%, SE=69.2863% After downscaling: S1=33.336%, S2=33.336%, SE=33.336% Total number of pulses: 66054



Look at correlation matrix

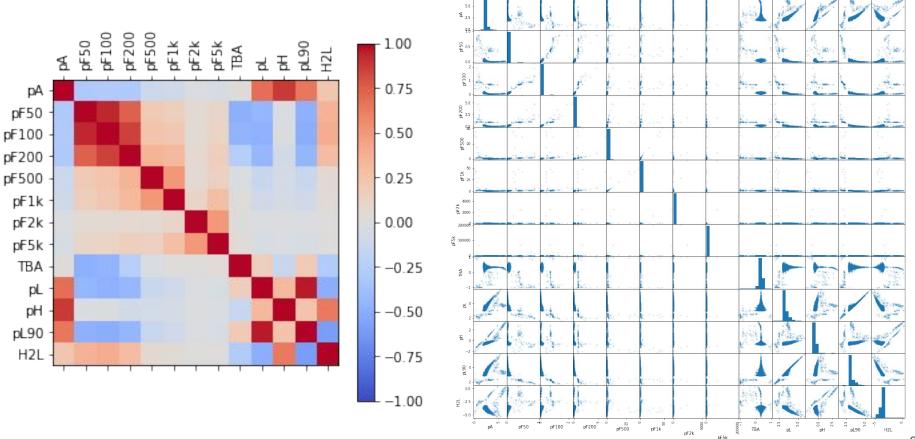


Combine features to mitigate correlation

```
# We can also apply non-linear transformations to the data.
#The scale of some features might span several orders of magnitude, so a log transformation might be useful
data_new = pd.DataFrame({
    'pA':np.log10(data_down['pA']),
    'pF50':data_down['pF50'],
    'pF100':(data_down['pA100']/data_down['pA']),
    'pF200':(data_down['pA200']/data_down['pA']),
    'pF500':(data_down['pA500']/data_down['pA']),
    'pF1k':(data_down['pA1k']/data_down['pA']),
    'pF2k':(data_down['pA2k']/data_down['pA']),
    'pF5k':(data_down['pA5k']/data_down['pA']),
    'TBA':data_down['TBA'],
    'pL':np.log10(data_down['pL']),
    'pH':np.log10(data_down['pH']),
    'pL90':np.log10((data_down['aft95']-data_down['aft5'])),
    'H2L':np.loq10((data_down['pH']/(data_down['aft95']-data_down['aft5'])))
```

})

Combine features to mitigate correlation



Scale data

```
# StandardScaler (the go-to and faithful) - sets mean to zero and stdv to 1
scaler = preprocessing.StandardScaler().fit(data_new)
data_scaled = scaler.transform(data_new)
data_scaled = pd.DataFrame(data_scaled)
data_scaled.columns = data_new.columns
```

We do not necessarily need scaling because we used a boosted tree based method.

Split dataset

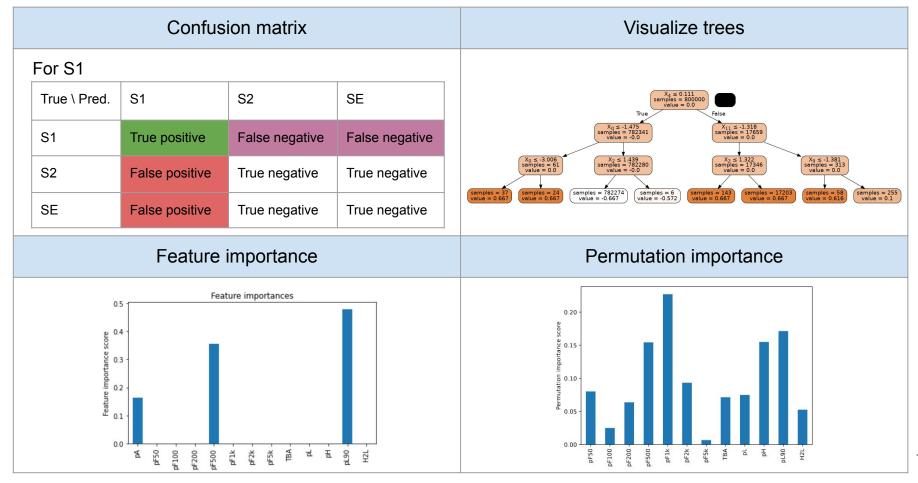
```
# Split dataset into training set and test set (80% training and 20% test)
X_train, X_test, y_train, y_test = train_test_split(data_scaled, labels_down, test_size=0.2, random_state=0)
print(len(X_train), len(y_test))
```

Training

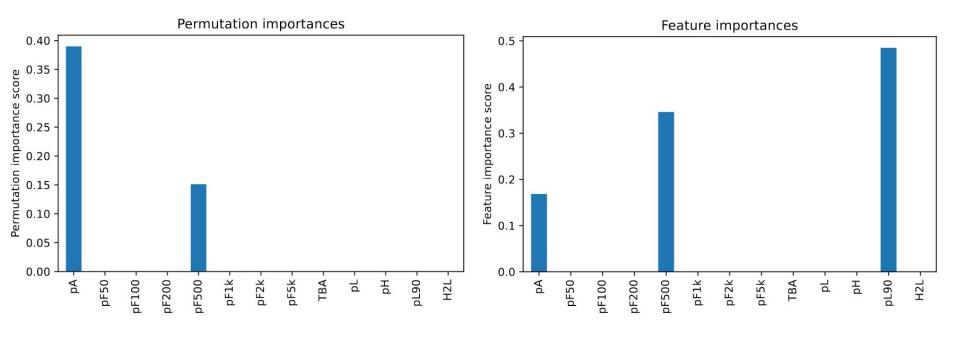
sklearn.ensemble.GradientBoostingClassifier

Parameters	Description	Default
max_depth	maximum depth of each tree	3
loss	loss function to be optimized	'deviance'
learning_rate	contribution of each tree	0.1
n_estimators	number of boosting stages	100
subsample	fraction of samples to be used for fitting	1
min_samples_split	minimum number of samples required to split an internal node	2
min_samples_leaf	minimum number of samples required to be at a leaf node	1

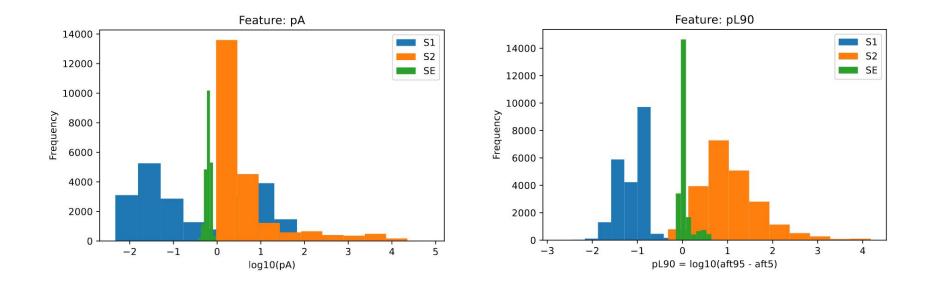
Training - How to evaluate the model?



Important features:

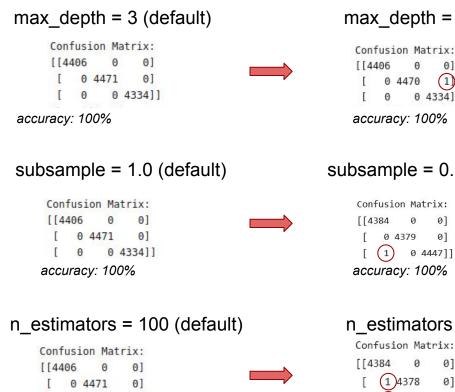


pulse area pulse fraction divided by pulse area pulse length of 90% of the pulse

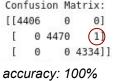


Confusion	Mat	rix:				٦	
[[4461	0	0]					
[0 434	46	0]					
[0	04	404]]					
Classifica	atic	on Report					
		precision	recall	f1-score	support		
	1	1.00	1.00	1.00	4461	}	too perfect
	2	1.00	1.00	1.00	4346		
	3	1.00	1.00	1.00	4404		
accura	асу			1.00	13211		
macro a	avg	1.00	1.00	1.00	13211		
weighted a	avg	1.00	1.00	1.00	13211		

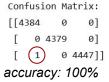
Model - Varying different hyperparameters



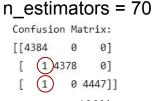
max depth = 10



subsample = 0.5



0 0 4334]] accuracy: 100%



accuracy: 100%

Preliminary result

Our model recreates the HADES tree.

Solution

Remove an important feature (e.g. pulse area or pulse fraction) to prevent our trees from recreating the HADES tree.

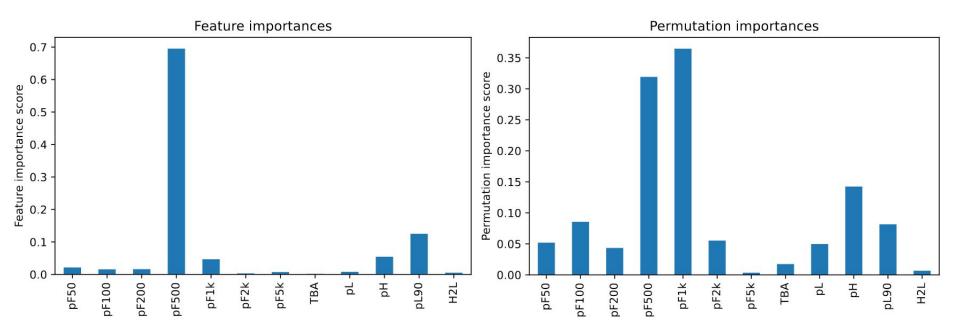
Final model - Removing pulse area

Confusion Matrix:		
[[A A A A A A A A A A A A A A A A A A A	max_depth = 5 #3	
[[4392 17 7]	loss = 'deviance'	
[21 4340 71]	learning_rate = 0.4 #0.	1
[6 66 4291]]	n_estimators = 40 #10	0
	subsample = 0.2 #1.	0
	<pre>max_sample_split = 10 #2</pre>	
Classification Depart	<pre>max_sample_leaf = 10 #1</pre>	

Classification Report

	precision	recall	f1-score	support
1	0.99	0.99	0.99	4416
2	0.98	0.98	0.98	4432
3	0.98	0.98	0.98	4363
accuracy			0.99	13211
macro avg	0.99	0.99	0.99	13211
weighted avg	0.99	0.99	0.99	13211

Final model - Removing pulse area



Conclusion

There was no overfitting of data but "overfitting of model" \rightarrow this shows that BDTs are extremely powerful at picking up nuances in the data \rightarrow finds the underlying model (HADES)

Solution: remove pulse area and use sufficient hyperparameters \rightarrow forces the model to fit the data and generalize it better

Results:

- ✓ model is no longer mimicking the HADES tree
- \checkmark most important features identified \rightarrow helps understand data better
- ✓ accuracy of 99%

Future work:

Use unsupervised learning, e.g. cluster analysis, in order to classify the data without using HADES



extra slides

Seperate labels from the data

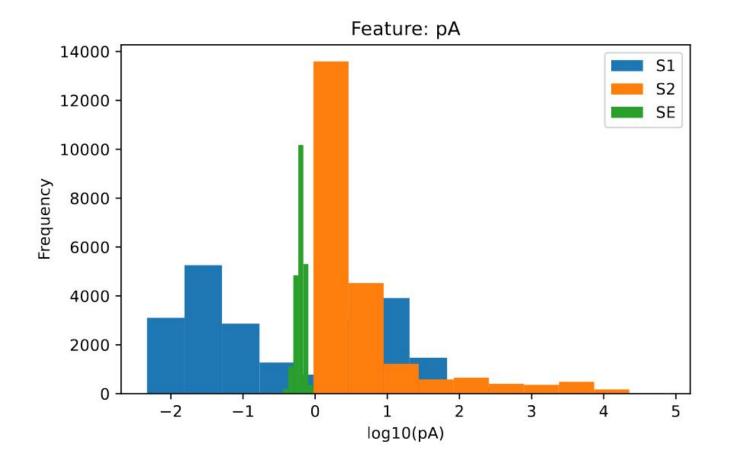
```
# separate labels from data for training
labels = pd.DataFrame(data, columns = ['pulseClass'])
data = data.drop('pulseClass', axis=1)
```

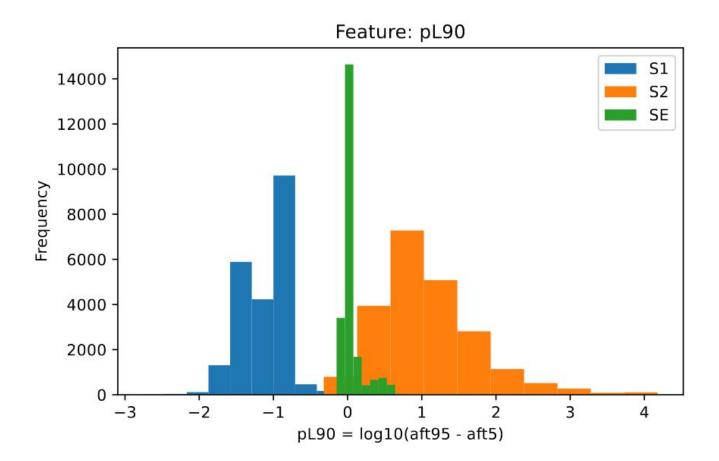
```
num_pulses = len(data)
print('Number of entries: {}'.format(num_pulses))
```

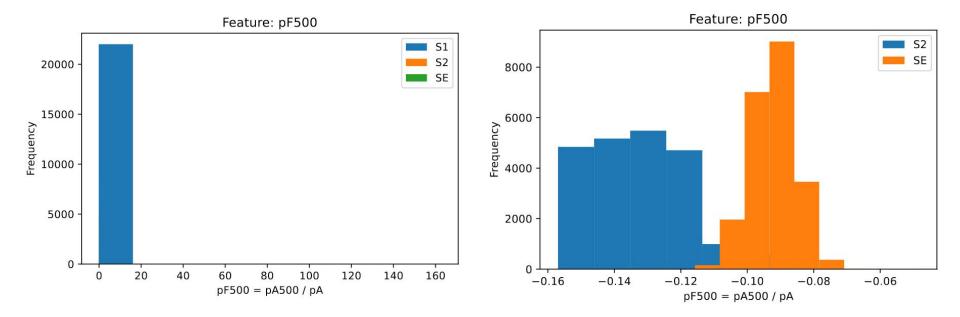
Number of entries: 1000000

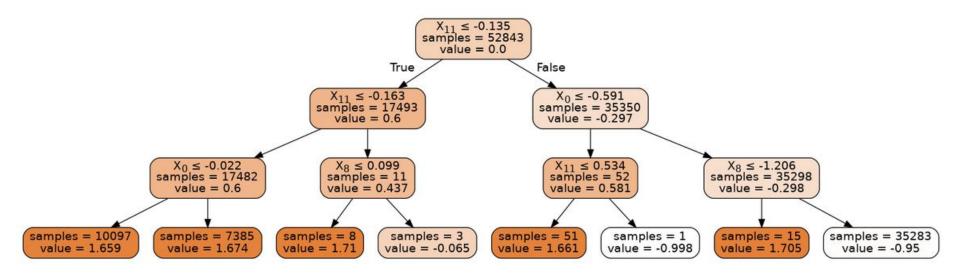
Combine features to improve correlation

```
# With some ingenuity the data features can be combined and improved (only valid if we understand our choices)
data_new = pd.DataFrame({
    'pA':data_down['pA'],
    'pF50':data_down['pF50'],
    'pF100':(data_down['pA100']/data_down['pA']),
    'pF200':(data_down['pA200']/data_down['pA']),
    'pF500':(data_down['pA500']/data_down['pA']),
    'pF1k':(data_down['pA1k']/data_down['pA']),
    'pF2k':(data_down['pA2k']/data_down['pA']),
    'pF5k':(data_down['pA5k']/data_down['pA']),
    'TBA':data_down['TBA'],
    'pL':data_down['pL'],
    'pH':data_down['pH'],
    'pL90':(data_down['aft95']-data_down['aft5']),
    'H2L':(data_down['pH']/(data_down['aft95']-data_down['aft5']))
})
```









Model 1 - Removing pulse area

<pre>md_value = 20 #3 l_value = 'deviance' lr_value = 0.4 #0.1 n_value = 40 #100 subs_value = 0.2 #1.0 mss_value = 35 #2 msl_value = 35 #1</pre>	Confusion Mat [[4402 6 [32 4298 [203 301 3 Classificatio	8] 102] 3859]]			
		precision	recall	f1-score	support
	1	0.95	1.00	0.97	4416
	2	0.93	0.97	0.95	4432
	3	0.97	0.88	0.93	4363
	accuracy			0.95	13211
	macro avg	0.95	0.95	0.95	13211
	weighted avg	0.95	0.95	0.95	13211

Model 2 - Removing pulse area

md_value = 5 #3						
l_value = 'deviance'	Confusion	Matrix:				
lr_value = 0.4 #0.1	[[4156 9	06 1641				
n_value = 40 #100						
subs_value = 0.2 #1.0	[0438	32 50]				
mss_value = 35 #2	[4 4	47 4312]]				
msl_value = 35 #1	Classifica	ation Rep	ort			
		prec	ision	recall	f1-score	support
		1	1.00	0.94	0.97	4416
		2	0.97	0.99	0.98	4432
		3	0.95	0.99	0.97	4363
	accura	асу			0.97	13211
	macro a	avg	0.97	0.97	0.97	13211
	weighted a	avg	0.97	0.97	0.97	13211

Seperate labels from the data

```
# separate labels from data for training
labels = pd.DataFrame(data, columns = ['pulseClass'])
data = data.drop('pulseClass', axis=1)
```

```
num_pulses = len(data)
print('Number of entries: {}'.format(num_pulses))
```

Number of entries: 1000000

Model - Varying different hyperparameters: max_depth

max_depth = 3 (defa	uitj				max_depth =	10				
Confusion Matr:	ix:				Confusion	Matrix	:			
[[4406 0	0]				[[4406	0 0				
	0]				[0 447		E)			
[0 0 433						0 4334				
Classification					Classifica	ation R	eport			
	precision	recall	fl-score	support		pr	ecision	recall	fl-score	support
1	1.00	1.00	1.00	4406		1	1.00	1.00	1.00	4406
2	1.00	1.00	1.00	4471		2	1.00	1.00	1.00	4471
3	1.00	1.00	1.00	4334		3	1.00	1.00	1.00	4334
accuracy			1.00	13211	accura	асу			1.00	13211
macro avg	1.00	1.00	1.00	13211	macroa	avg	1.00	1.00	1.00	13211
weighted avg	1.00	1.00	1.00	13211	weighted a	pvg	1.00	1.00	1.00	13211
nergineed dvg										
					econdertas = ectificador		2-0-2-0.0-0/02		4 10 10 10 10 Kes	
max_depth = 5	x:				max_depth = 2		20000000		Tradicial An	
max_depth = 5 Confusion Matri					econdertas = ectificador		2990900200			
max_depth = 5 Confusion Matri [[4406 0	0]				max_depth = 2		2960930960			
max_depth = 5 Confusion Matri [[4406 0 [0 4470	0] 1]				max_depth = 2 Confusion Matrix: [[4406 0 0] [0 4471 0]	2	20000000			
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433	0] 1] 4]]				max_depth 2 Confusion Matrix: [[4406 0 0] [0 4471 0] [[0 0 4334]] [2	240400000			
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433 Classification	0] 1] 4]] Report	recall	fl.scoro	support	max_depth = 2 Confusion Matrix: [[4406 0 0] [0 4471 0]	2	74043040			
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433 Classification	0] 1] 4]]	recall	f1-score	support	max_depth 2 Confusion Matrix: [[4406 0 0] [0 4471 0] [[0 0 4334]] Classification Rep	2	recall	f1-score	e support	
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433 Classification	0] 1] 4]] Report	recall 1.00	f1-score 1.00	support 4406	max_depth = 2 Confusion Matrix: [[4406 0 0] [0 4471 0] [0 0 4334]] Classification Rep prec	2 ort ision				
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433 Classification P	0] 1] 4]] Report Precision				max_depth = 2 Confusion Matrix: [[4406 0 0] [0 4471 0] [0 0 4334]] Classification Rep prec 1	2 ort ision 1.00	1.00	1.00) 4406	
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433 Classification P 1	0] 1] 4]] Report precision 1.00	1.00	1.00	4406	max_depth = 2 Confusion Matrix: [[4406 0 0] [0 4471 0] [0 0 4334]] Classification Rep prec	2 ort ision) 4406) 4471	
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433 Classification p 1 2	0] 1] 44]] Report precision 1.00 1.00	1.00 1.00	1.00	4406 4471	max_depth = 2 Confusion Matrix: [[4406 0 0] [0 4471 0] [0 0 4334]] Classification Rep prec 1 2 3	2 ort ision 1.00 1.00	1.00 1.00	1.00 1.00 1.00) 4406) 4471) 4334	
max_depth = 5 Confusion Matri [[4406 0 [0 4470 [0 0 433 Classification p 1 2 3	0] 1] 44]] Report precision 1.00 1.00	1.00 1.00	1.00 1.00 1.00	4406 4471 4334	max_depth = 2 Confusion Matrix: [[4406 0 0] [0 4471 0] [0 0 4334]] Classification Rep prec 1 2	2 ort ision 1.00 1.00	1.00 1.00	1.00) 4406) 4471) 4334) 13211	

Model - Varying different hyperparameters: learning_rate

	(defaul	,			learning_rate = 0.8		
Confusion Matri	x:				Confusion Matrix:		
[[4406 0	0]				[[4406 0 0]		
[0 4471	0]				[0 4471 0]		
[0 0 433	4]]				[0 0 4334]]		
Classification	Report				Classification Report		
	recision	recall	fl-score	support	precision re	all f1-scor	e support
1	1.00	1.00	1.00	4406	1 1.00	00 1.0	0 4406
2	1.00	1.00	1.00	4471	2 1.00	00 1.0	0 4471
3	1.00	1.00	1.00	4334	3 1.00	00 1.0	0 4334
accuracy			1.00	13211	accuracy	1.0	0 13211
macro avg	1.00	1.00	1.00	13211	macro avg 1.00	.00 1.0	0 13211
weighted avg	1.00	1.00	1.00	13211	weighted avg 1.00	.00 1.0	0 13211

learning_rate =	0.3				
Confusion	Matri	x:			
[[4406	Θ	0]			
[0 44]	71	Θ]			
0]	0 433	34]]			
Classifica	ation	Report			
	F	precision	recall	fl-score	support
	1	1.00	1.00	1.00	4406
	2	1.00	1.00	1.00	4471
	3	1.00	1.00	1.00	4334
accura	асу			1.00	13211
macro a	avg	1.00	1.00	1.00	13211
weighted a	avg	1.00	1.00	1.00	13211

learning_rate = 0.9 and max_depth = 1

Confusior	n Mat	rix:			
[[4406	0	0]			
[044	171	0]			
[0	0 43	334]]			
Classific	atio	n Report			
		precision	recall	fl-score	support
	1	1.00	1.00	1.00	4406
	2	1.00	1.00	1.00	4471
	3	1.00	1.00	1.00	4334
accur	асу			1.00	13211
macro	avg	1.00	1.00	1.00	13211
weighted	avg	1.00	1.00	1.00	13211

Model - Varying different hyperparameters: n_estimators

n_estimators = 100 (default)	n_estimators = 10
Confusion Matrix: [[4406 0 0]	Confusion Matrix: [[4384 0 0]
[0 4471 0] [0 0 4334]] Classification Report	[1 4378 0] [0 0 4448]] Classification Report
precision recall f1-score suppo	
1 1.00 1.00 1.00 44 2 1.00 1.00 1.00 44	
3 1.00 1.00 1.00 43	2 1100 1100 4575
accuracy 1.00 132 macro avg 1.00 1.00 1.00 132 weighted avg 1.00 1.00 1.00 132	11 macro avg 1.00 1.00 13211
n_estimators = 40	n_estimators = 70

<pre>[[4384 0 0] [0 4379 0] [1 0 4447]] Classification Report</pre>	Confusion Mat	trix:			
<pre>[1 0 4447]] Classification Report</pre>	[[4384 0	0]			
Classification Report precision recall f1-score support 1 1.00 1.00 1.00 4384 2 1.00 1.00 1.00 4379 3 1.00 1.00 1.00 4448	[0 4379	0]			
precision recall f1-score support 1 1.00 1.00 1.00 4384 2 1.00 1.00 1.00 4379 3 1.00 1.00 1.00 4448	[1 04	4447]]			
1 1.00 1.00 1.00 4384 2 1.00 1.00 1.00 4379 3 1.00 1.00 1.00 4448	Classificatio	on Report			
2 1.00 1.00 1.00 4379 3 1.00 1.00 1.00 4448		precision	recall	f1-score	support
2 1.00 1.00 1.00 4379 3 1.00 1.00 1.00 4448					
3 1.00 1.00 1.00 4448	1	1.00	1.00	1.00	4384
	2	1.00	1.00	1.00	4379
	3	1.00	1.00	1.00	4448
accuracy 1.00 13211	accuracy			1.00	13211
macro avg 1.00 1.00 1.00 13211	macro avg	1.00	1.00	1.00	13211
weighted avg 1.00 1.00 1.00 13211	weighted avg	1.00	1.00	1.00	13211

Confusion M	atrix:			
[[4384 0	0]			
[1 4378	0]			
[1 0	4447]]			
Classificat	ion Report			
	precision	recall	f1-score	support
	1 1.00	1.00	1.00	4384
	2 1.00	1.00	1.00	4379
	3 1.00	1.00	1.00	4448
accurac	y		1.00	13211
macro av	g 1.00	1.00	1.00	13211
weighted av	g 1.00	1.00	1.00	13211

Model - Varying different hyperparameters: subsample

subsample = 1 (default)		subsample = 0.2						
Confusion Matrix:		Confusion Matrix:						
[[4406 0 0]	[[4384 0 0	9]						
[0 4471 0]	[0 4379 0]							
	$\begin{bmatrix} 1 & 0 & 4447 \end{bmatrix}$							
Classification Report	Classification F							
	f1-score s	upport			all f1.	-score sup	nort	
precision recati	TI-Score S	upport	P.		urr ir	Score Sup		
1 1.00 1.00	1.00	4406	1	1.00	1.00	1.00	4384	
2 1.00 1.00	1.00	4471	2	1.00	1.00	1.00	4379	
3 1.00 1.00	1.00	4334	3	1.00	L.00	1.00	4448	
accuracy	1.00	13211	accuracy			1.00 1	3211	
macro avg 1.00 1.00	1.00	13211	macro avg	1.00	1.00		3211	
weighted avg 1.00 1.00	1.00	13211	weighted avg		1.00		.3211	
subsample = 0.5	subsample = 0.8							
			Confusion Ma	trix:				
Confusion Matrix:			[[4384 0	0]				
[[4384 0 0]			[0 4379	0]				
[0 4379 0]	[1 0 4447]]							
[1 0 4447]]	Classification Report							
Classification Report				precision	recall	f1-score	support	
precision recall f1	-score support			PLECTOTON	recall	11-30016	Support	
1 1.00 1.00	1.00 4384		1	1.00	1.00	1.00	4384	
2 1.00 1.00	1.00 4379		2	1.00	1.00	1.00	4379	

3

accuracy

macro avg

weighted avg

1.00

1.00

1.00

1.00

1.00

1.00

1.00

1.00

1.00

1.00

4448

13211

13211

13211

1.00

1.00

1.00

1.00

1.00

1.00

3

accuracy

macro avg

weighted avg

1.00

1.00

1.00

1.00

4448

13211

13211

13211

Model - Varying different hyperparameters: min_samples_split

mples_split =	•					min_samples_split = 10				
Confusion Matri					Confusion Mat					
[[4406 0 0]					[[4384 0	0]				
[0 4471 0]					[1 4378	0]				
[0 0 4334]]					E 1000 100 1	447]]				
	Classification Report			Classificatio						
F	recision	recall	f1-score	support		precision	recall	f1-score	support	
1	1.00	1.00	1.00	4406	1	1.00	1.00	1.00	4384	
2	1.00	1.00	1.00	4471	2	1.00	1.00	1.00	4379	
3	1.00	1.00	1.00	4334	3	1.00	1.00	1.00	4448	
accuracy			1.00	13211	accuracy			1.00	13211	
macro avg	1.00	1.00	1.00	13211	macro avg	1.00	1.00	1.00	13211	
weighted avg	1.00	1.00	1.00	13211			1.00		13211	
mples split =	20				weighted avg	1.00 solit = :		1.00	13211	
mples_split =					min_samples_	split = :		1.00	13211	
Confusion Matr	ix:				min_samples_ Confusion M	split = (1.00	13211	
Confusion Matr [[4384 0	ix: 0]				min_samples_ confusion M [[4384 6	split = (1.00	13211	
Confusion Matr [[4384 0 [1 4378	ix: 0] 0]				min_samples_ confusion M [[4384 @ [1 4378	split = (1.00	15211	
Confusion Matr [[4384 0	ix: 0] 0] 47]]				min_samples_ Confusion M [[4384 0 [1 4378 [1 6	split = :		1.00		
Confusion Matr [[4384 0 [1 4378 [1 0 44 Classification	ix: 0] 0] 47]]	recall	f1-score	support	min_samples_ confusion M [[4384 @ [1 4378	split = :	30	1.00		
Confusion Matr [[4384 0 [1 4378 [1 0 44 Classification	ix: 0] 0] 47]] Report	recall	f1-score 1.00	support 4384	min_samples_ Confusion N [[4384 0 [1 4378 [1 6 Classificat	split = 3 Natrix: 0 0] 0 4447]] cion Report precisio	30 n recal	l f1-score	suppor	
Confusion Matr [[4384 0 [1 4378 [1 0 44 Classification	ix: 0] 0] 47]] Report precision				min_samples_ Confusion N [[4384 0 [1 4378 [1 6 Classificat	<pre>split = : hatrix: 0] 4447]] cion Report precisio 1 1.0</pre>	30 n recal ø 1.0	l f1-score 0 1.00	suppor 4384	
Confusion Matr [[4384 0 [1 4378 [1 0 44 Classification 1	ix: 0] 0] 47]] Report precision 1.00	1.00	1.00	4384	min_samples_ Confusion N [[4384 0 [1 4378 [1 6 Classificat	<pre>split = : atrix: 0] 4447]] ion Report precisio 1 1.0 2 1.0</pre>	30 n recal 0 1.0 0 1.0	l f1-score 0 1.00 0 1.00	support 438/ 4379	
Confusion Matr [[4384 0 [1 4378 [1 0 44 Classification 1 2	ix: 0] 0] 47]] Report precision 1.00 1.00	1.00	1.00 1.00	4384 4379	min_samples_ Confusion N [[4384 0 [1 4378 [1 6 Classificat	<pre>split = : hatrix: 0] 4447]] cion Report precisio 1 1.0</pre>	30 n recal 0 1.0 0 1.0	l f1-score 0 1.00 0 1.00	suppor 438 437	
Confusion Matr [[4384 0 [1 4378 [1 0 44 Classification 1 2	ix: 0] 0] 47]] Report precision 1.00 1.00	1.00	1.00 1.00	4384 4379	min_samples_ Confusion N [[4384 0 [1 4378 [1 6 Classificat	<pre>split = : hatrix: 0] 0] 4447]] ion Report precisio 1 1.0 2 1.0 3 1.0</pre>	30 n recal 0 1.0 0 1.0	l f1-score 0 1.00 0 1.00	support 4384 4379 4448	
Confusion Matr [[4384 0 [1 4378 [1 0 44 Classification 1 2 3	ix: 0] 0] 47]] Report precision 1.00 1.00	1.00	1.00 1.00 1.00	4384 4379 4448	min_samples_ Confusion M [[4384 0 [1 4374 [1 4374 [1 6 Classificat	<pre>split = : hatrix: 0] 0] 4447]] ion Report precisio 1 1.0 2 1.0 3 1.0</pre>	30 n recal 0 1.0 0 1.0 0 1.0	l f1-score 0 1.00 0 1.00 0 1.00 1.00	support 4384 4375 4441 13212	

Model - Varying different hyperparameters: min_samples_leaf

_samples_leaf = 1 (default)						min_samples_split = 5					
Confusion Matrix:						Confusion Matrix:					
[[4406 0 0]						[[4384 0	0]				
[0 4471 0]						[1 4378 0]					
[0 0 4334]]						[1 0 4447]]					
Classification Report						Classification	Report				
ſ	orecision	recall	fl-score	support		ţ	precision	recall	f1-score	support	
1	1.00	1.00	1.00	4406		1	1.00	1.00	1.00	4384	
2	1.00	1.00	1.00	4471		2	1.00	1.00	1.00	4379	
3	1.00	1.00	1.00	4334		3	1.00	1.00	1.00	4448	
accuracy			1.00	13211		accuracy			1.00	13211	
macro avg	1.00	1.00	1.00	13211		macro avg	1.00	1.00	1.00	13211	
weighted avg	1.00	1.00	1.00	13211		weighted avg	1.00	1.00	1.00	13211	
	10				min oor		1-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0				
[1 4378 [1 0 444 Classification	x: 0] 0] 7]]	recall f	1-score s	upport	min_sar	mples_split = Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification	= 20 x: 0] 0] 7]]		f1-score		
Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification	x: 0] 0] 7]] Report	recall f 1.00	1-score s 1.00	upport 4384	min_sar	nples_split = Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification [= 20 x: 0] 0] 77]] Report precision 1.00	recall 1.00	f1-score 1.00	support 4384	
Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification F	x: 0] 0] 7]] Report precision				min_sar	mples_split = Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification 1 2	= 20 x: 0] 0] 77]] Report orecision 1.00 1.00	recall 1.00 1.00	f1-score 1.00 1.00	support 4384 4379	
Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification F	x: 0] 0] 7]] Report precision 1.00	1.00	1.00	4384	min_sar	nples_split = Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification [= 20 x: 0] 0] 77]] Report precision 1.00	recall 1.00	f1-score 1.00	support 4384	
Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification F 1 2	x: 0] 0] 7]] Report recision 1.00 1.00	1.00 1.00	1.00 1.00	4384 4379	min_sar	mples_split = Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification 1 2	= 20 x: 0] 0] 77]] Report orecision 1.00 1.00	recall 1.00 1.00	f1-score 1.00 1.00	support 4384 4379	
Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification F 1 2 3	x: 0] 0] 7]] Report recision 1.00 1.00	1.00 1.00	1.00 1.00 1.00	4384 4379 4448	min_sar	mples_split = Confusion Matri [[4384 0 [1 4378 [1 0 444 Classification 1 2 3	= 20 x: 0] 0] 77]] Report orecision 1.00 1.00	recall 1.00 1.00	f1-score 1.00 1.00 1.00	support 4384 4379 4448	

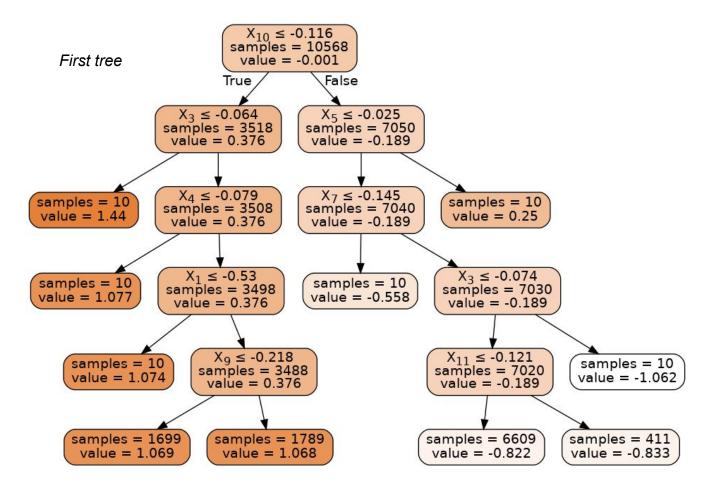
The data

Load the data for this challenge
filename = 'Data/Data_Challenge_1.csv'

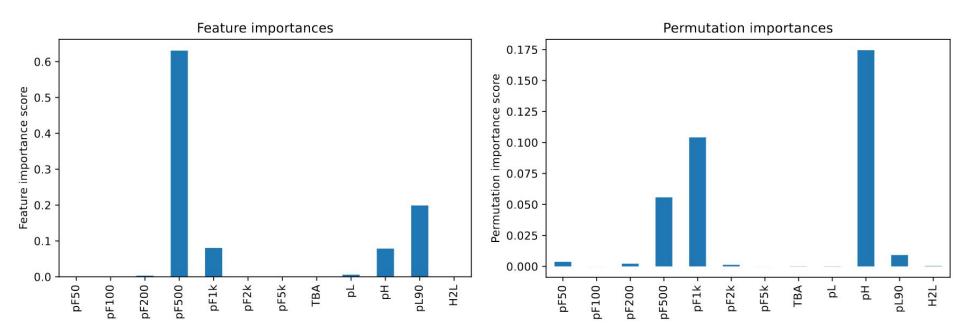
```
data = pd.read_csv(filename)
data.describe()
```

						Preview	🝷 🔤 🗠 Visualize
	pulseClass flo… 🔤	pA float64 🛛 🔤	pF50 float64 🔤	TBA float64 🔤	pH float64 🛛 🖾	pL float64 🛛 🔛	р∟90 тіоать4 🔤
count	1000000	1000000	1000000	1000000	1000000	1000000	1000000
mean	2.670845	14805.921483824 119	0.0480317010443 7695	0.3232947484664 078	11.400557485129 177	5056.75925	4077.62274
std	0.5146340941127 501	174136.07271127 097	0.0760875023503 9809	0.1701743724081 0382	141.38568024706 163	26457.410470036 706	23443.104537888 86
min	1	0.800442695618	0.000006284829	-2.473263502121	0.003778859042	20	10
25%	2	110.16364479064 95	0.024427488446	0.2766844183202 5004	0.2275129668412 5	1180	830
50%	3	124.34373855590 8	0.0386589895935 00004	0.3380941897635 0003	0.2593486607074 9996	1470	890
75%	3	224.79640197753 9	0.0523605206985	0.3960270434617 5	0.3067244887352 5	3060	1930
max	3	22276876	3.558622837067	2.716481924057	12126.434570312 5	795440	701980
<							>

Final model - Removing pulse area



Model - Removing pulse area (with default hyperparam.)



Model - Removing pulse area (with default hyperparam.)

Confusion Mat	rix:				٦	
[[4416 0	0]					
[0 4402	30]					
[0 24 4	339]]					
Classificatio	on Report					
	precision	recall	f1-score	support		
						better but still
1	1.00	1.00	1.00	4416	7	too perfect
2	0.99	0.99	0.99	4432		
3	0.99	0.99	0.99	4363		
accuracy			1.00	13211		
macro avg	1.00	1.00	1.00	13211		
weighted avg	1.00	1.00	1.00	13211	J	

try different hyperparameters values until the model confusion matrix looks "realistic"