LZ pulse classification

TAAD - Final Project

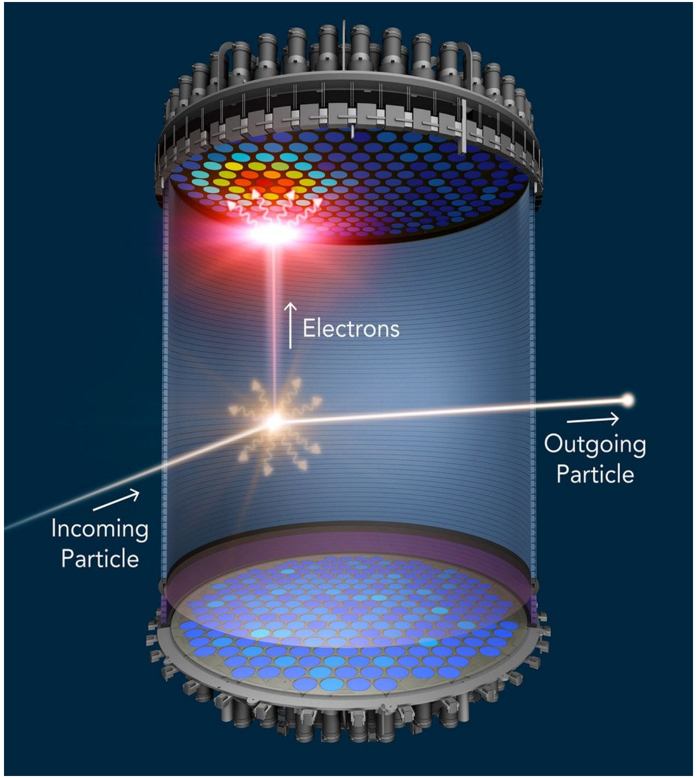
04.02.2022

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

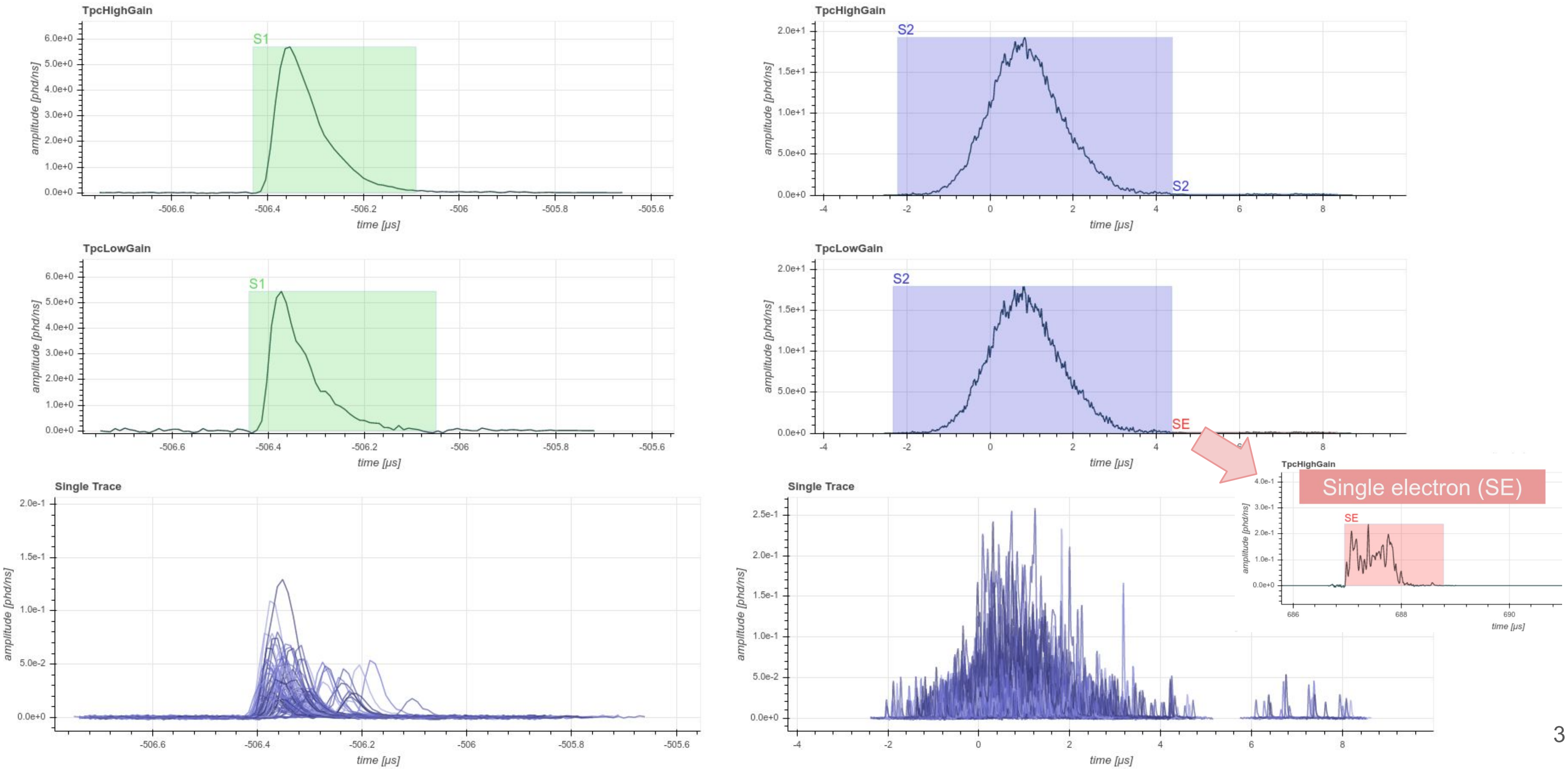
The LZ is a dark matter direct detection experiment expected to start running in 2022. It will use a two-phase xenon Time Projection Chamber (TPC) to detect Weakly Interacting Massive Particles (WIMPs).



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

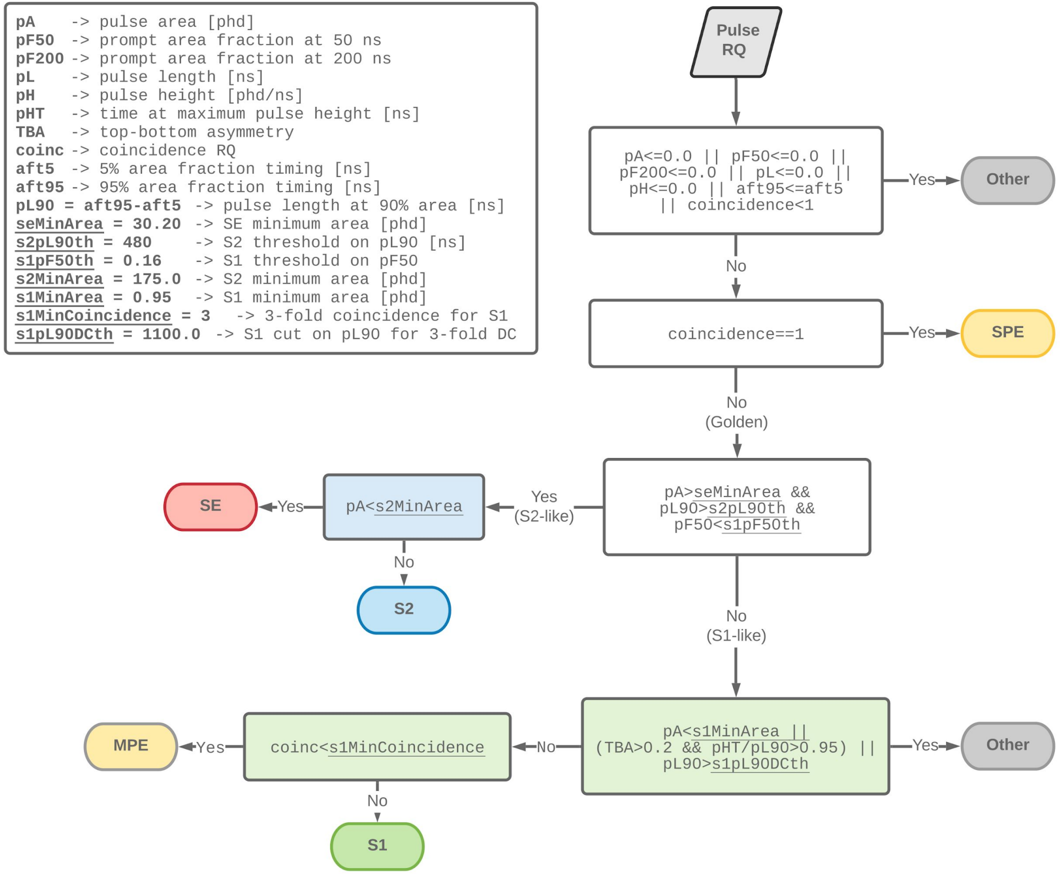


Current pulse classifier

HADES (Heuristic Algorithm for Discrimination of Event Substructures)

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

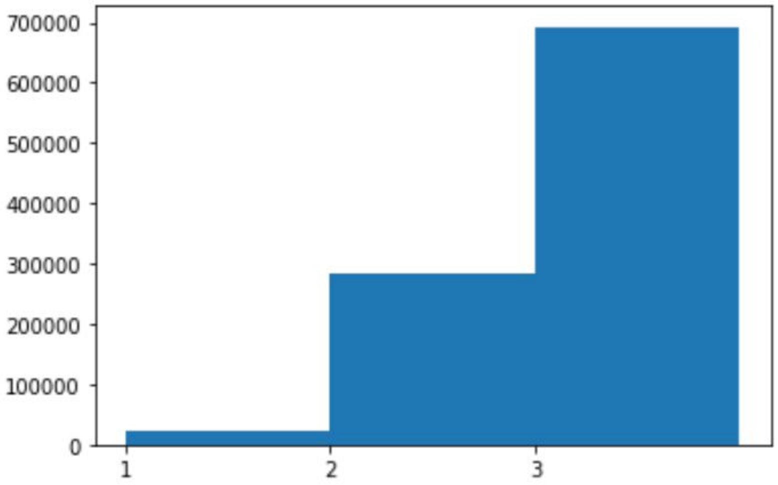
Estimated classification accuracy: 98.6%



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 '''
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)
```

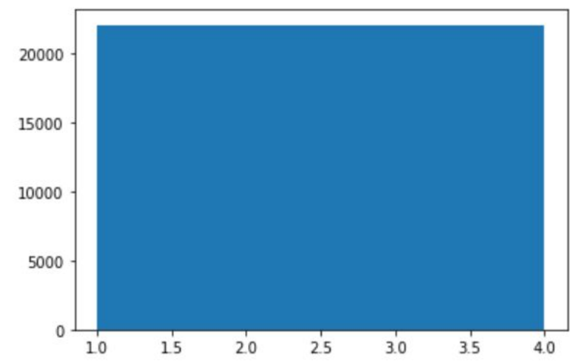


S1=2.2018%, S2=28.5119%, SE=69.2863%

under-represented

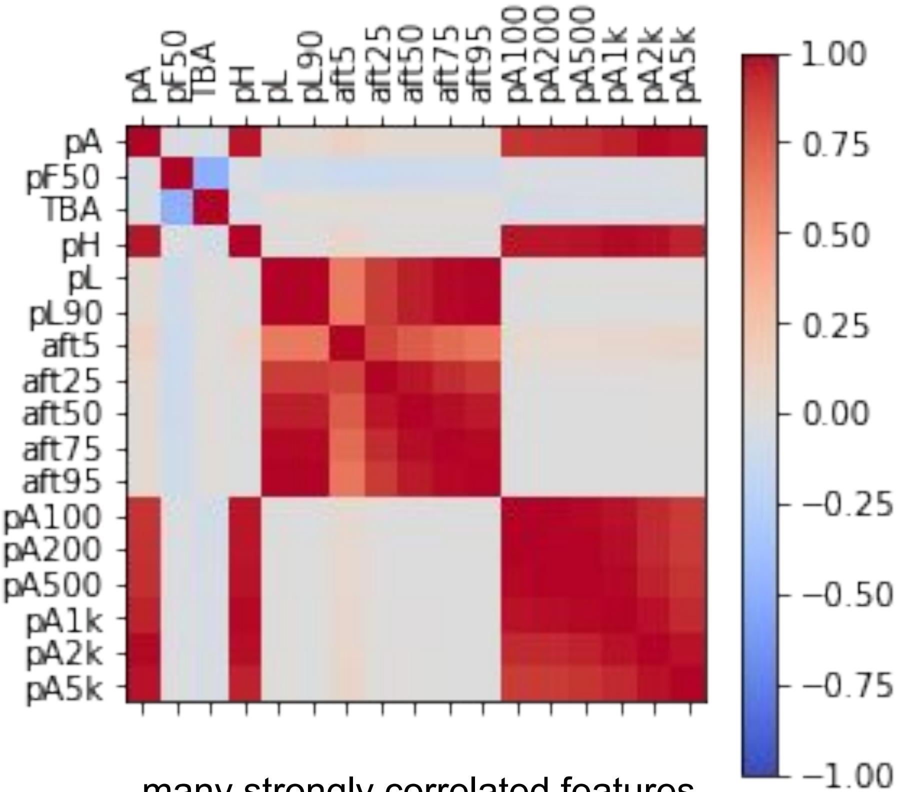
Scale down data

```
''' Scale down data because S2 and S3 are over represented '''
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={}%, SE={}%'.format(100*len(S1)/num_pulses, 100*len(S2)/num_pulses, 100*len(SE)/n

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])
```

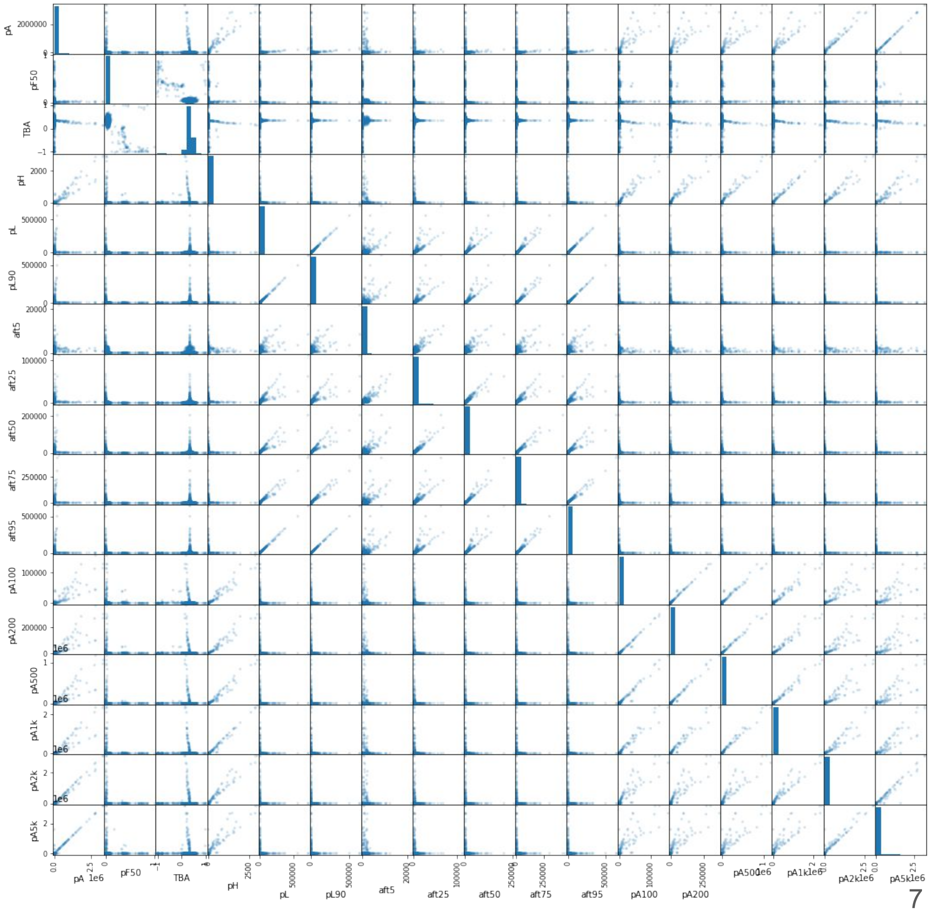


Before downscaling: S1=2.2018%, S2=28.5119%, SE=69.2863%
After downscaling: S1=33.336%, S2=33.336%, SE=33.336%
Total number of pulses: 66054

Look at correlation matrix



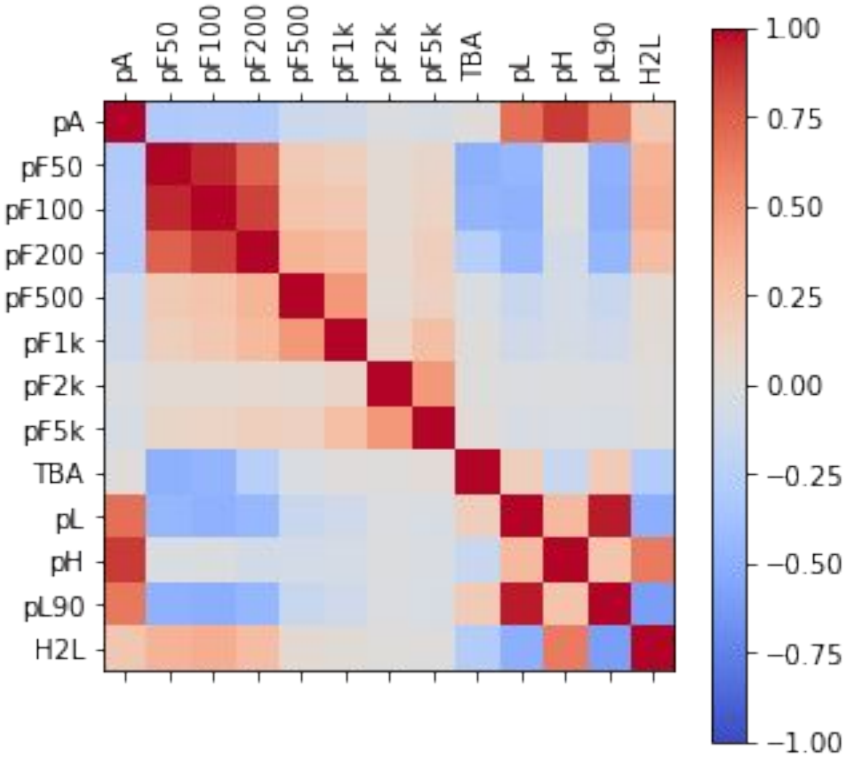
many strongly correlated features

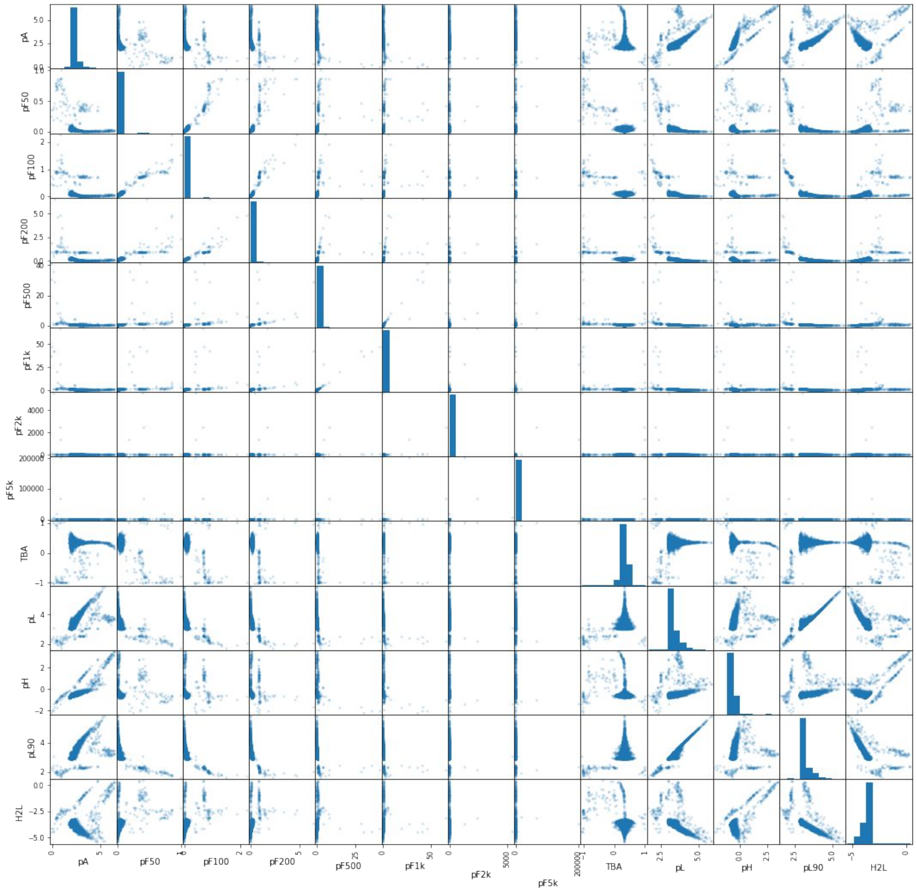


Combine features to mitigate correlation

```
# 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.log10((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

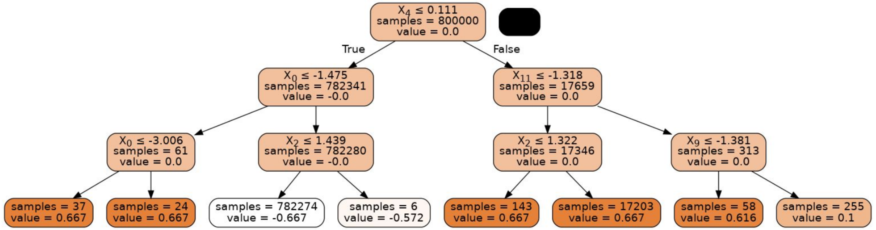
Training - How to evaluate the model?

Confusion matrix

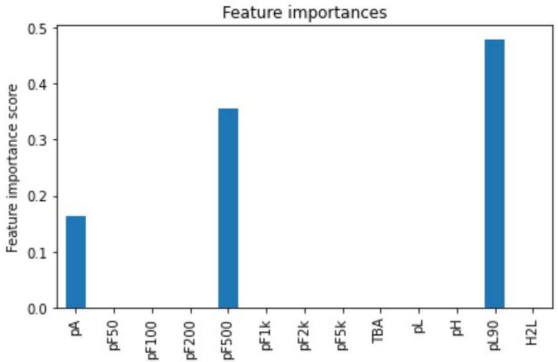
For S1

True \ Pred.	S1	S2	SE
S1	True positive	False negative	False negative
S2	False positive	True negative	True negative
SE	False positive	True negative	True negative

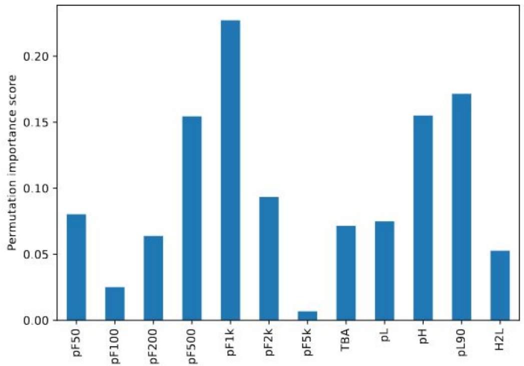
Visualize trees



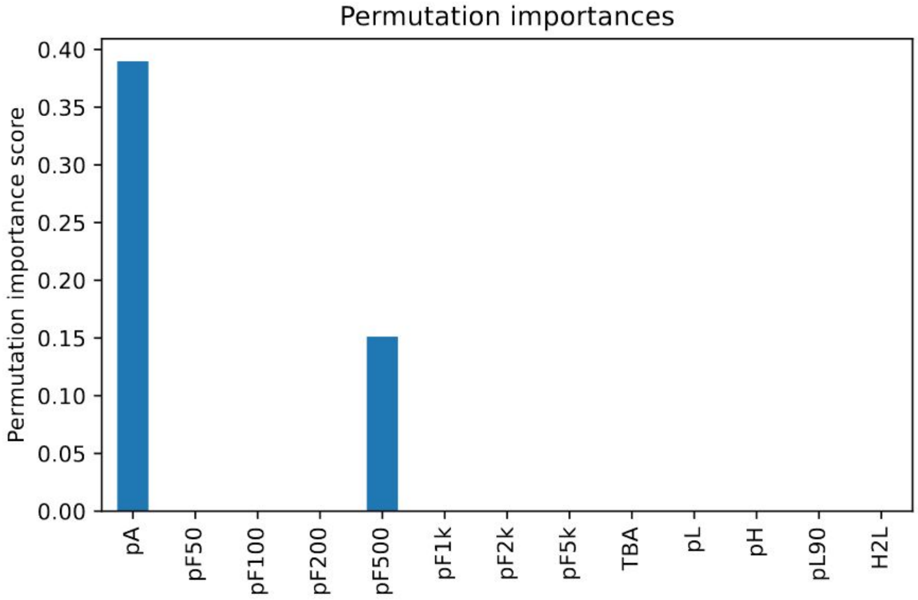
Feature importance

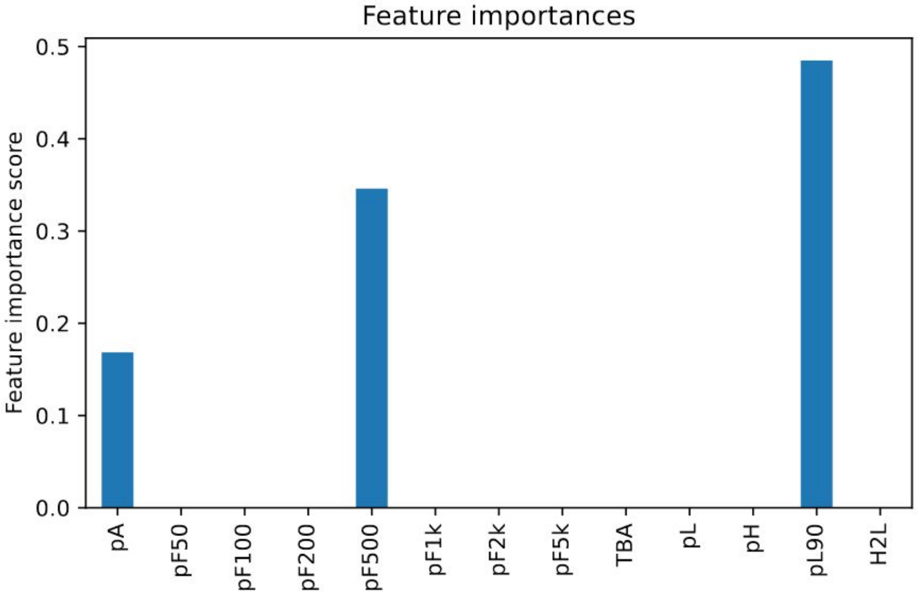


Permutation importance



Model - With default hyperparameters

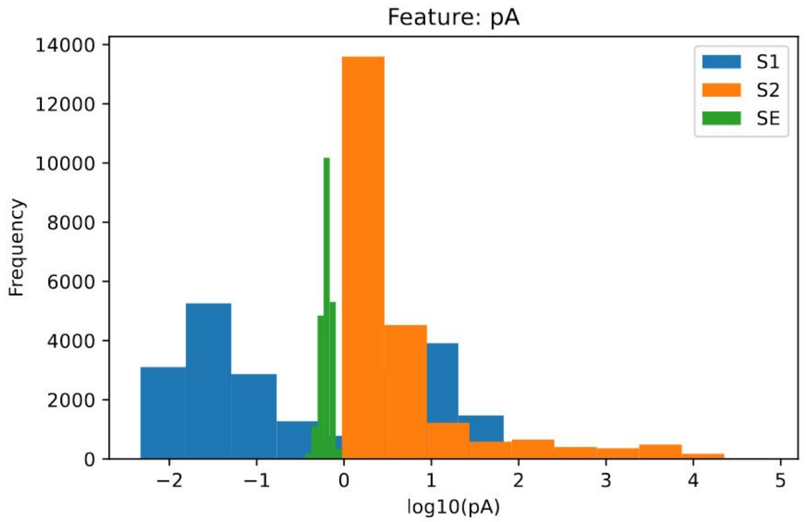


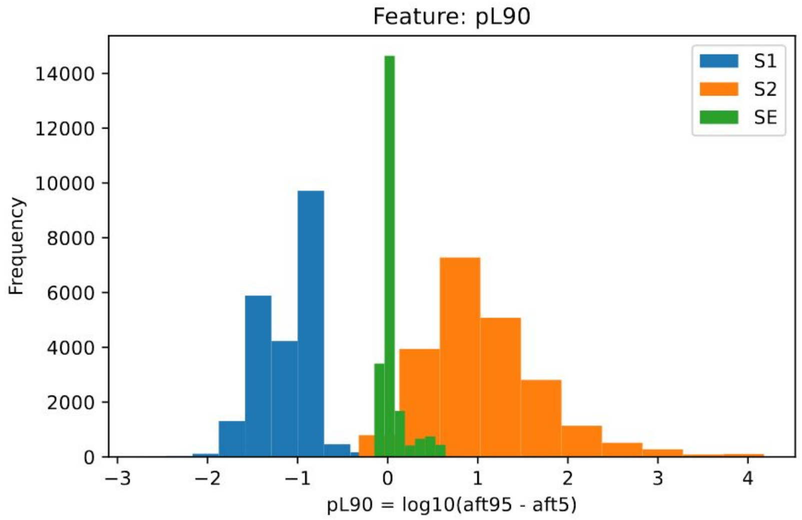


Important features:

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

Model - With default hyperparameters





Model - With default hyperparameters

```
Confusion Matrix:
[[4461    0    0]
 [   0 4346    0]
 [   0    0 4404]]
Classification Report
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      4461
           2       1.00      1.00      1.00      4346
           3       1.00      1.00      1.00      4404

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

too perfect

Model - Varying different hyperparameters

max_depth = 3 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
```

accuracy: 100%

max_depth = 10

```
Confusion Matrix:
[[4406    0    0]
 [   0 4470    1]
 [   0    0 4334]]
```

accuracy: 100%

subsample = 1.0 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
```

accuracy: 100%

subsample = 0.5

```
Confusion Matrix:
[[4384    0    0]
 [   0 4379    0]
 [   1    0 4447]]
```

accuracy: 100%

n_estimators = 100 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
```

accuracy: 100%

n_estimators = 70

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
```

accuracy: 100%

Preliminary result

Our model recreates the HADES tree.

Solution

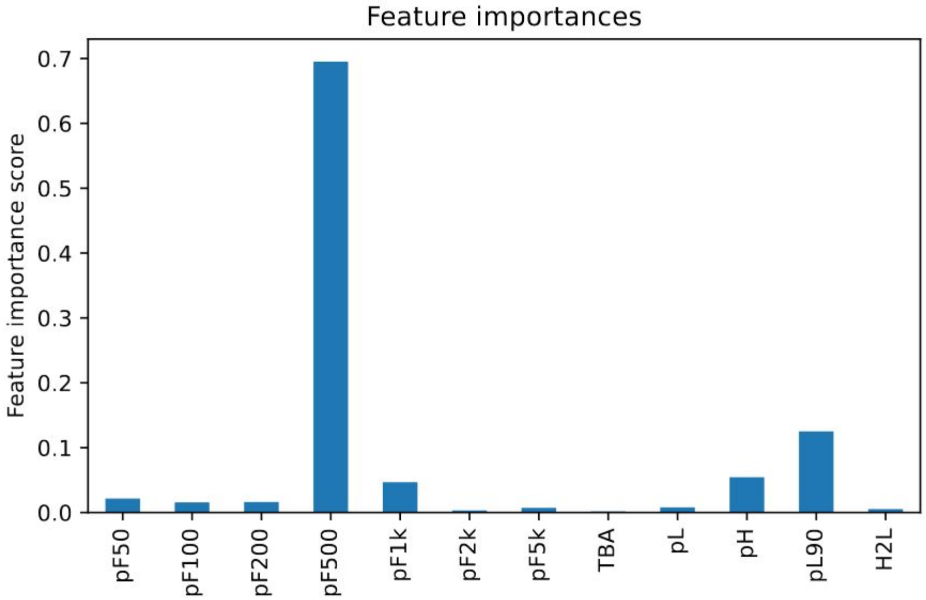
Remove an important feature (e.g. pulse area or pulse fraction) to prevent our trees from recreating the HADES tree.

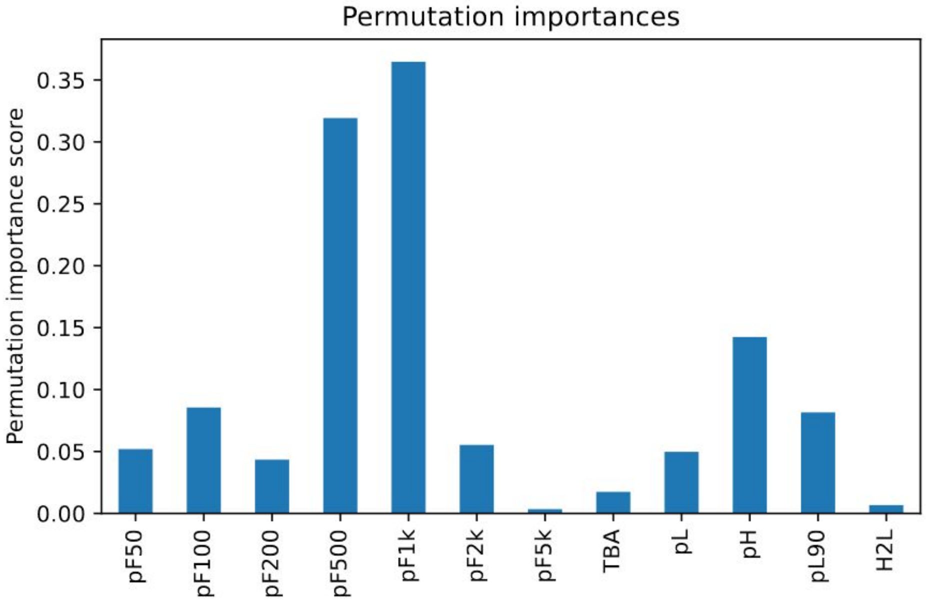
Final model - Removing pulse area

```
Confusion Matrix:
[[4392   17    7]
 [  21 4340   71]
 [   6   66 4291]]

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
```

```
max_depth =  5            #3
loss = 'deviance'
learning_rate = 0.4       #0.1
n_estimators = 40         #100
subsample = 0.2           #1.0
max_sample_split = 10     #2
max_sample_leaf = 10      #1
```

Final model - Removing pulse area





Conclusion

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

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

Results:

- ✓ model is no longer mimicking the HADES tree
- ✓ most important features identified → 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

Thank you for your attention!

Any questions?

extra slides

Seperate labels from the data

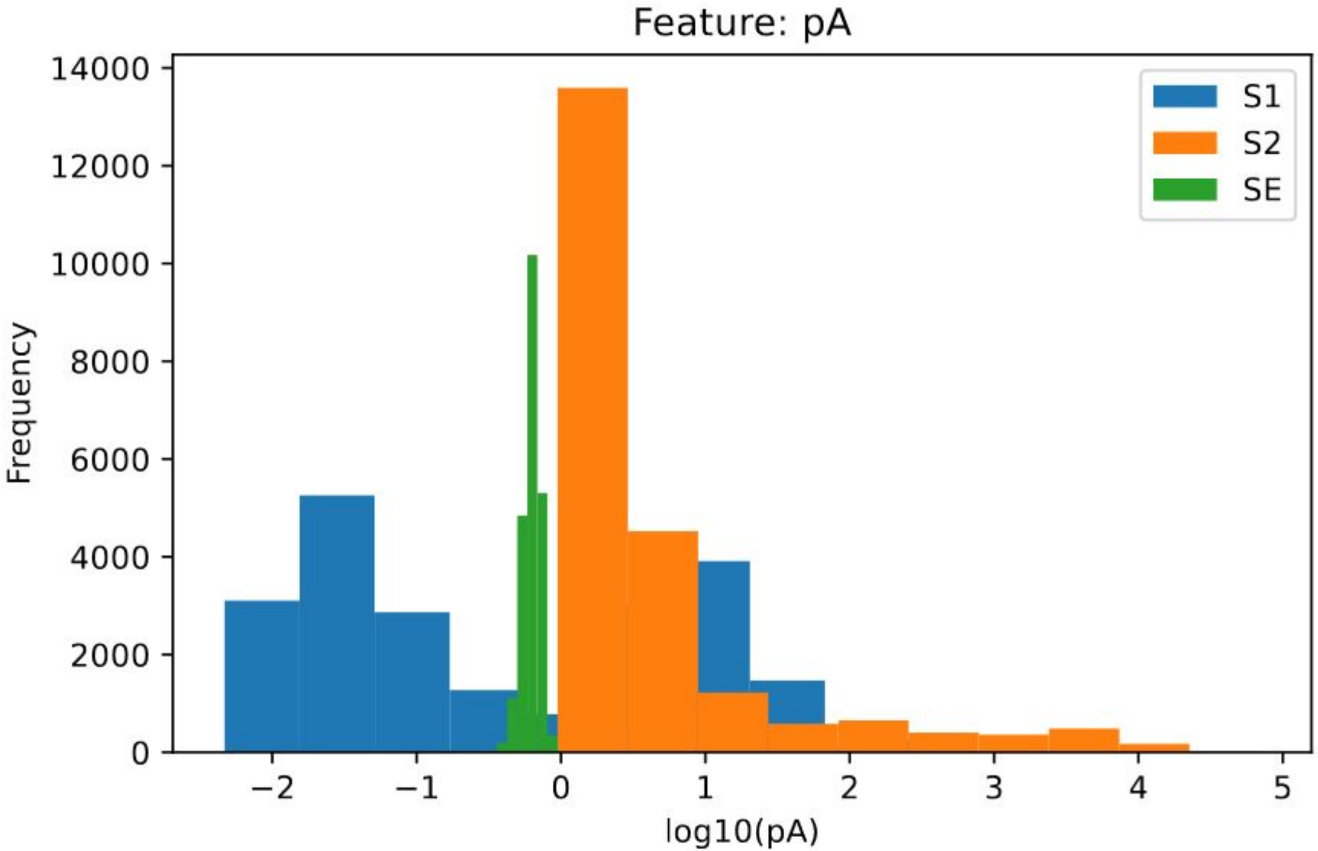
```
# separate labels from data for training
labels = pd.DataFrame(data, columns = ['pulseClass'])
data = data.drop('pulseClass', axis=1)

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

```
Number of entries: 1000000
```

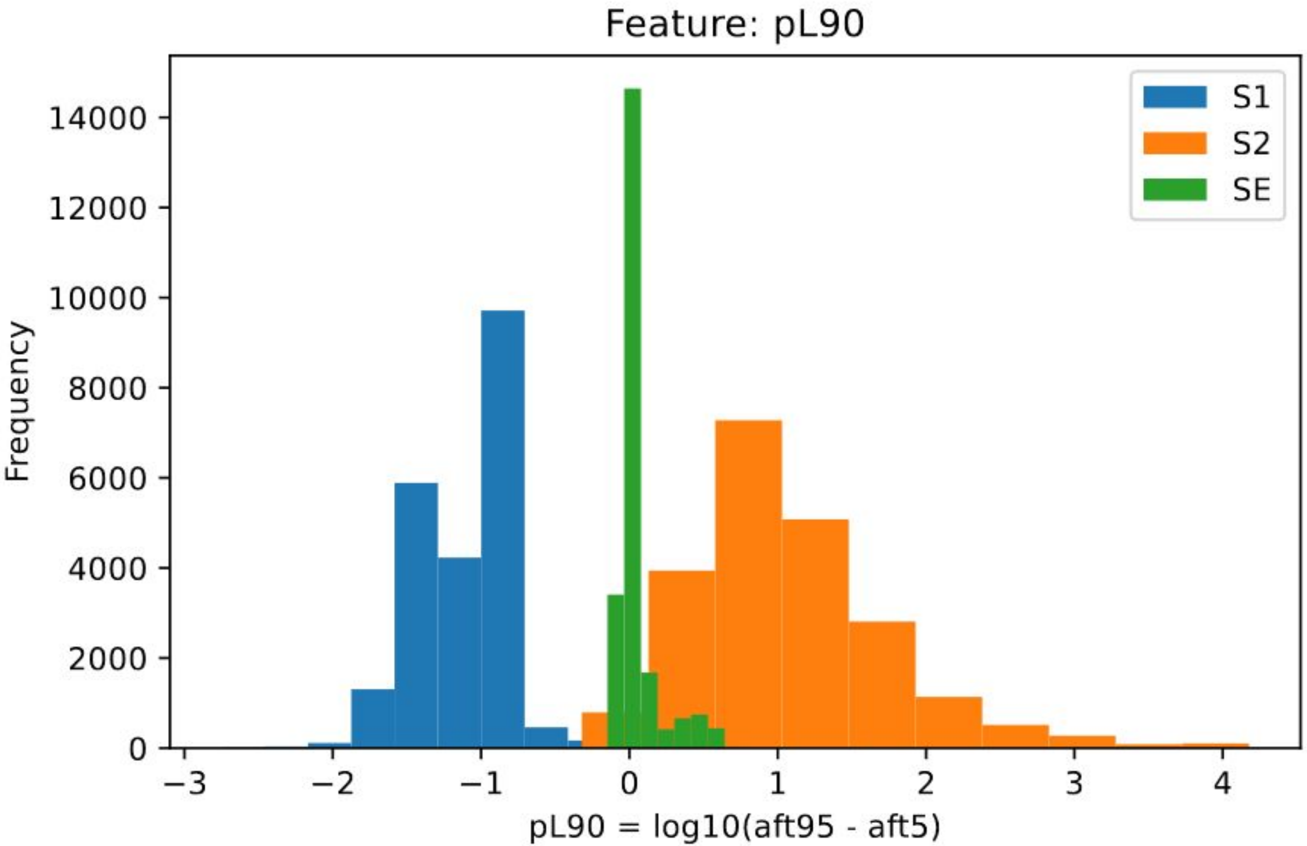
Combine features to improve correlation

```
# 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']))
})
```

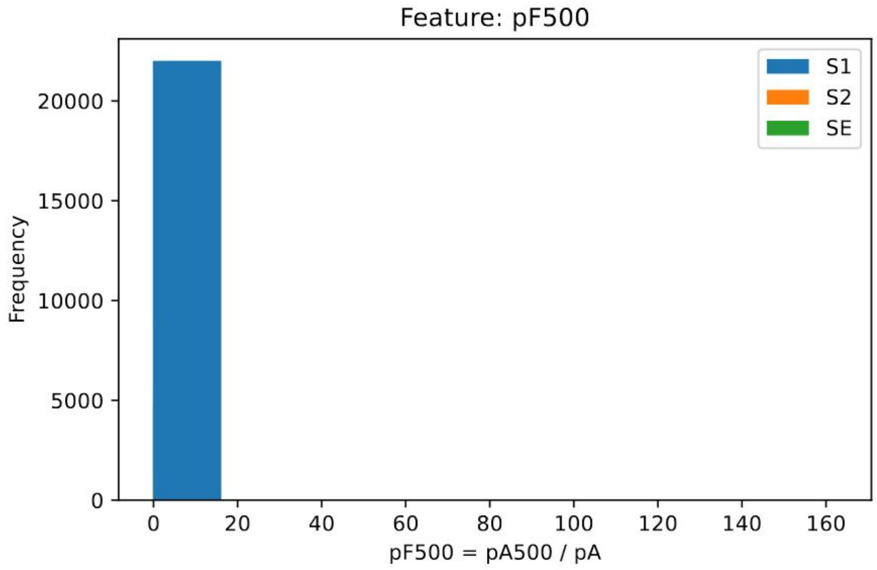
Model - With default hyperparameters

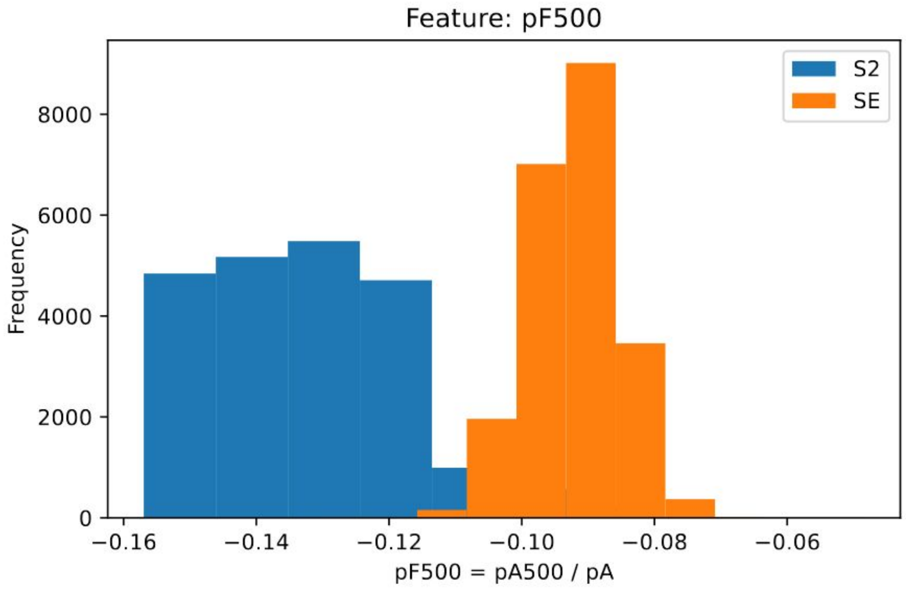


Model - With default hyperparameters

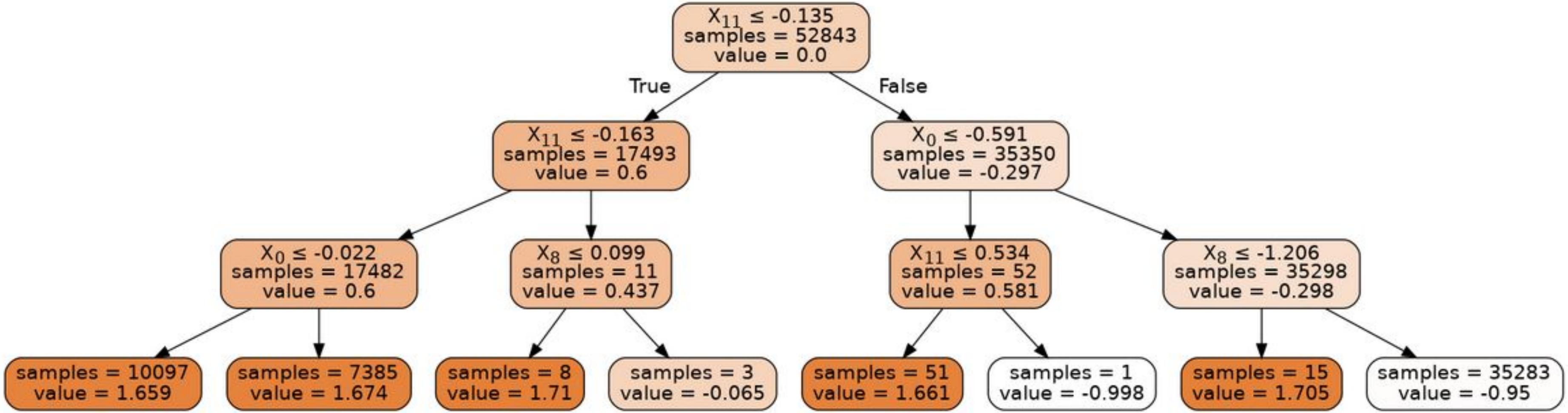


Model - With default hyperparameters





Model - With default hyperparameters



Model 1 - Removing pulse area

```
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
```

```
Confusion Matrix:
[[4402    6    8]
 [  32 4298  102]
 [ 203  301 3859]]
Classification Report
              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'
lr_value = 0.4 #0.1
n_value = 40 #100
subs_value = 0.2 #1.0
mss_value = 35 #2
msl_value = 35 #1
```

```
Confusion Matrix:
[[4156   96  164]
 [   0 4382   50]
 [   4   47 4312]]
Classification Report
              precision    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

    accuracy                           0.97     13211
   macro avg       0.97      0.97      0.97     13211
weighted 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 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

max_depth = 10

```
Confusion Matrix:
[[4406    0    0]
 [   0 4470    1]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

max_depth = 5

```
Confusion Matrix:
[[4406    0    0]
 [   0 4470    1]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

max_depth = 2

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           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: learning_rate

learning_rate = 0.1 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

learning_rate = 0.8

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

learning_rate = 0.3

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

learning_rate = 0.9 and max_depth = 1

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           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)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

n_estimators = 10

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   0    0 4448]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

n_estimators = 40

```
Confusion Matrix:
[[4384    0    0]
 [   0 4379    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

n_estimators = 70

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
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

    accuracy                           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: subsample

subsample = 1 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

subsample = 0.2

```
Confusion Matrix:
[[4384    0    0]
 [   0 4379    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

subsample = 0.5

```
Confusion Matrix:
[[4384    0    0]
 [   0 4379    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

subsample = 0.8

```
Confusion Matrix:
[[4384    0    0]
 [   0 4379    0]
 [   1    0 4447]]
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

    accuracy                           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: min_samples_split

min_samples_split = 2 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

min_samples_split = 10

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

min_samples_split = 20

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

min_samples_split = 30

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
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

    accuracy                           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: min_samples_leaf

min_samples_leaf = 1 (default)

```
Confusion Matrix:
[[4406    0    0]
 [   0 4471    0]
 [   0    0 4334]]
Classification Report
              precision    recall  f1-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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

min_samples_split = 5

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

min_samples_split = 10

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

min_samples_split = 20

```
Confusion Matrix:
[[4384    0    0]
 [   1 4378    0]
 [   1    0 4447]]
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

    accuracy                           1.00     13211
   macro avg       1.00      1.00      1.00     13211
weighted avg       1.00      1.00      1.00     13211
```

The data

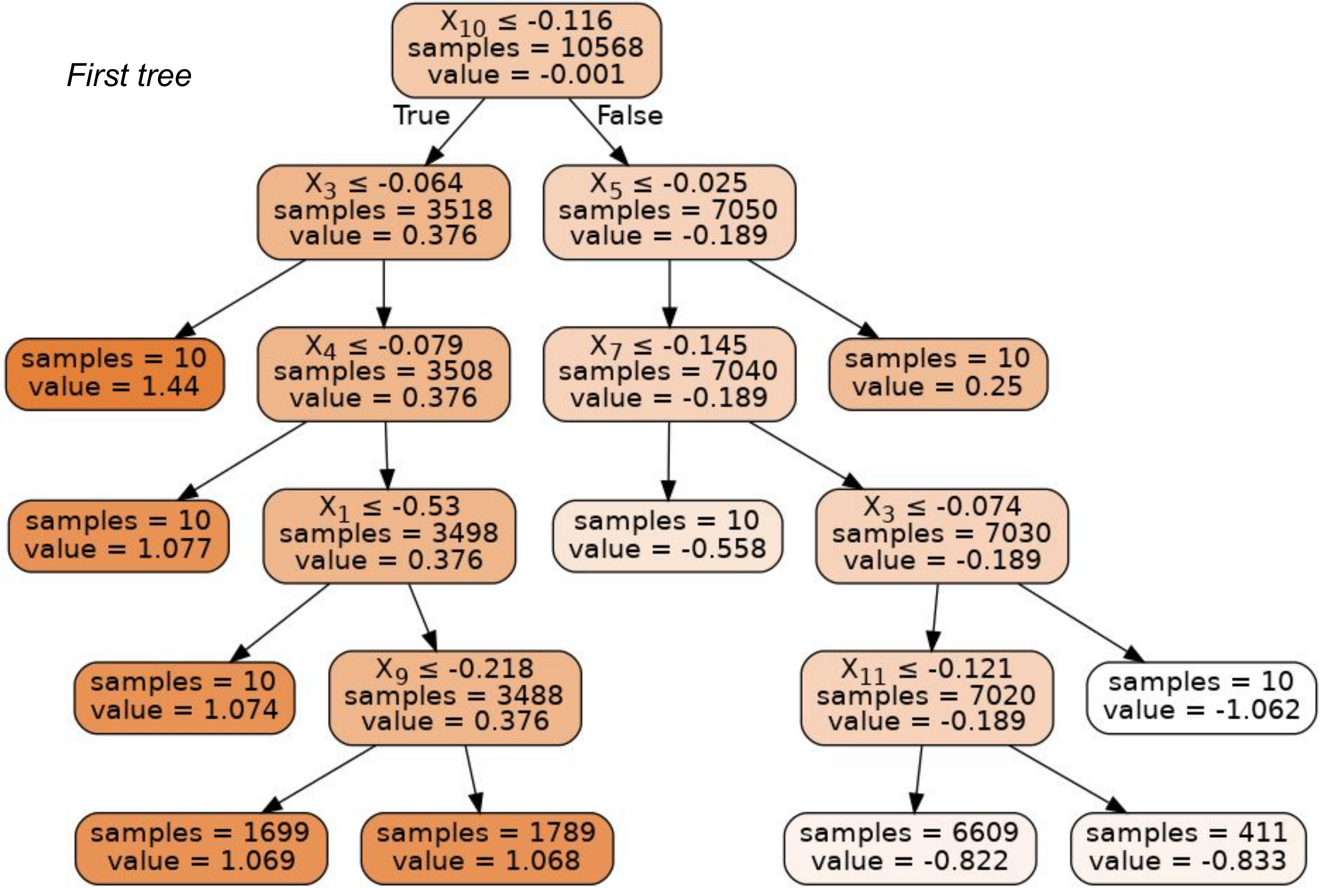
```
# Load the data for this challenge
filename = 'Data/Data_Challenge_1.csv'

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

Preview Visualize

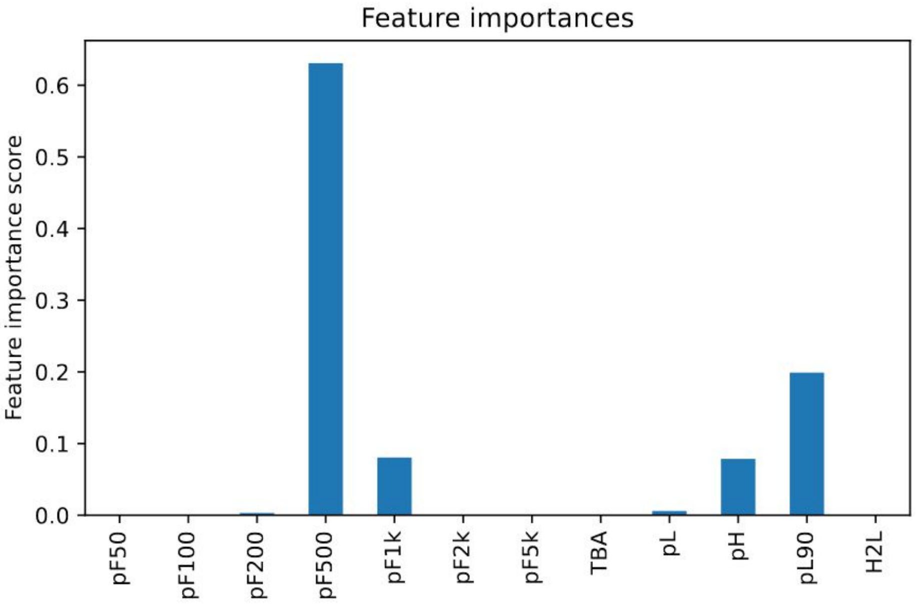
	pulseClass flo…	pA float64	pF50 float64	TBA float64	pH float64	pL float64	pL90 float64
count	1000000	1000000	1000000	1000000	1000000	1000000	1000000
mean	2.670845	14805.921483824119	0.04803170104437695	0.3232947484664078	11.400557485129177	5056.75925	4077.62274
std	0.5146340941127501	174136.07271127097	0.07608750235039809	0.17017437240810382	141.38568024706163	26457.410470036706	23443.10453788886
min	1	0.800442695618	0.000006284829	-2.473263502121	0.003778859042	20	10
25%	2	110.1636447906495	0.024427488446	0.27668441832025004	0.22751296684125	1180	830
50%	3	124.343738555908	0.038658989593500004	0.33809418976350003	0.25934866070749996	1470	890
75%	3	224.796401977539	0.0523605206985	0.39602704346175	0.30672448873525	3060	1930
max	3	22276876	3.558622837067	2.716481924057	12126.4345703125	795440	701980

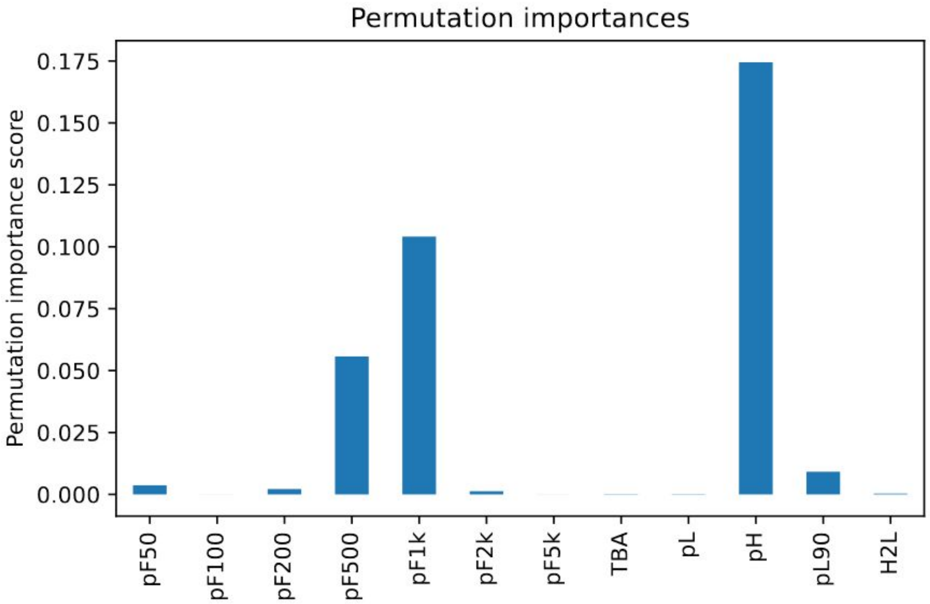
Final model - Removing pulse area

First tree



Model - Removing pulse area (with default hyperparam.)





Model - Removing pulse area (with default hyperparam.)

```
Confusion Matrix:
[[4416    0    0]
 [   0 4402   30]
 [   0   24 4339]]
Classification Report
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      4416
           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
```

better but still too perfect

try different hyperparameters values until the model confusion matrix looks “realistic”